



CHEDIS TER

RH133

Red Hat Linux System Administration

RH133-RHEL5-en-1-20070321

redhat®

**GLOBAL LEARNING SERVICES
TRAINING AND CERTIFICATION PROGRAMS**

Lesia Chaytan

rh 133

Aug 11, 2017

log M - root/red hnt

pw - Red stem

VALUE OF A RED HAT SUBSCRIPTION

Stable. Secure. Scalable.

Red Hat subscriptions provide enterprise-ready solutions that can be confidently deployed to manage even your most mission-critical applications.

SUBSCRIPTION BENEFITS

- **KEEP YOUR SYSTEMS SECURE** - Get the latest, extensively tested security patches through Red Hat Network, our content delivery and systems management tool.
- **ACCESS SUPPORT YOU CAN TRUST** - Deploy confidently with the backing of Red Hat experts. Each release is supported for seven years.
- **AUTOMATE EVERYDAY MAINTENANCE TASKS** - Save time and become more efficient with additional Red Hat Network capabilities.
- **GET THE LATEST SOFTWARE FASTER** - Access newly tested updates and versions.
- **KEEP YOUR TECHNOLOGY UNDER CONTROL** - Simplify managing certifications and version control.
- **GET BETTER PERFORMANCE AND SUPPORT** - Improve performance. Systems running the latest updates will perform better and make issue-resolution easier.
- **PROTECT YOUR INVESTMENT** - Spread costs and value over time, and protect your technology with Red Hat's Open Source Assurance program, our promise to replace software if there is an intellectual property issue.
- **GROW YOUR TECHNOLOGY INFRASTRUCTURE RESPONSIBLY**
 - Gain greater control over your investment.

Subscriptions are an easier, more efficient way of doing business.

> Learn more about the value of your subscription: redhat.com/explore/value



Table of Contents

RH133 - Red Hat Linux System Administration

RH133 Red Hat Enterprise Linux Administration

Copyright	ix
Welcome	x
Participant Introductions	xi
Red Hat Enterprise Linux	xii
Red Hat Enterprise Linux Variants	xiii
Red Hat Network	xiv
Other Red Hat Supported Software	xv
Notes on Internationalization	xvi
The Fedora Project	xvii
Classroom Network	xviii
Objectives	xix
Audience and Prerequisites	xx

Unit 1 - System Initialization

Objectives	2
Boot Sequence Overview	3
Boot Loader Components	4
GRUB and grub.conf	5
Starting the Boot Process: GRUB	6
Kernel Initialization	7
init Initialization	8
Run Levels	9
/etc/rc.d/rc.sysinit	10
/etc/rc.d/rc	11
System V run levels	12
/etc/rc.d/rc.local	13
Controlling Services	14
End of Unit 1	15
Lab 1: Managing Startup	16
Sequence 1: Changing the default run level	17

Unit 2 - Package Management

Objectives	20
RPM Package Manager	21
Installing and Removing Software	22
Updating a Kernel RPM	23
rpm Queries	24
rpm Verification	25

About yum	26
Using yum	27
Searching packages/files	28
Configuring Additional Repositories	29
Creating a private repository	30
Red Hat Network	31
Red Hat Network Server	32
Entitlements	33
Red Hat Network Client	34
End of Unit 2	35
Lab 2: Working with packages	36
Sequence 1: Using RPM	37
Sequence 2: Connecting to a private repository	39
Sequence 3: Installing new packages using yum	40
Sequence 4: Updating software using yum	41

Unit 3 - Kernel Services

Objectives	48
The Linux Kernel	49
Kernel Images and Variants	50
Kernel Modules	51
Kernel Module Utilities	52
Managing the initrd Image	54
Accessing Drivers Through /dev	55
Device Node Examples	56
Managing /dev With udev	57
Adding Files Under /dev	58
Kernel Configuration With /proc	59
/proc Examples	60
sysctl : Persistent Kernel Configuration	61
Exploring Hardware Devices	62
Monitoring Processes and Resources	63
Lab 3: Configuring the kernel	64
Sequence 1: Turning off ping responses	65
Optional Sequence 2: Creating a file persistently under /dev/	66
Sequence 3: Exploring processes, hardware and memory resources	67

Unit 4 - System Services

Objectives	72
Network Time Protocol	73
System Logging	74
syslog Configuration	75
XOrg: The X11 Server	76
XOrg Server Configuration	78
XOrg in runlevel 3	79

XOrg in runlevel 5	80
Remote X Sessions	81
SSH: Secure Shell	82
VNC: Virtual Network Computing	83
cron	84
Controlling Access to cron	85
System crontab Files	86
Daily Cron Jobs	87
The anacron System	88
CUPS	89
End of Unit 4	90
Lab 4: System Services	91
Sequence 1: Using cron	92
Sequence 2: Logging to a centralized loghost	93
Sequence 3: Setting up a printer and administering a printer with CUPS.	94

Unit 5 - User Administration

Objectives	99
Adding a New User Account	100
User Private Groups	101
Modifying / Deleting User Accounts	102
Group Administration	103
Password Aging Policies	104
Switching Accounts	105
sudo	106
Network Users	107
Authentication Configuration	108
Example: NIS Configuration	109
Example: LDAP Configuration	110
SUID and SGID Executables	111
SGID Directories	112
The Sticky Bit	113
Default File Permissions	114
Access Control Lists (ACLs)	115
SELinux	117
SELinux, continued	118
SELinux: Targeted Policy	120
SELinux: Management	121
End of Unit 5	122
Lab 5: User and Group Administration	123
Sequence 1: Creating the groups and users	124
Sequence 2: Setting up shared directories	125
Sequence 3: Access Control Lists	126
Sequence 4: Understanding Security Context	127
Sequence 5: Client-side NIS account management	128
Sequence 6: Client-side LDAP account management	129

Unit 6 - Filesystem Management

Objectives	141
Overview: Adding New Filesystems to the Filesystem Tree	142
Device Recognition	143
Disk Partitioning	144
Managing Partitions	145
Making Filesystems	146
Filesystem Labels	147
tune2fs	148
Mount Points and <code>/etc/fstab</code>	149
Mounting Filesystems with mount	151
Unmounting Filesystems	152
mount By Example	153
Handling Swap Files and Partitions	154
Mounting NFS Filesystems	155
Automounter	156
Direct Maps	158
gnome-mount	159
End of Unit 6	160
Lab 6: Adding New Filesystems to the Filesystem Tree	161
Sequence 1: Creating and Mounting Filesystems	162
Sequence 2: Mounting an NFS Filesystem	163
Sequence 3: Automounting data with autofs	164

Unit 7 - Advanced Filesystem Management

Objectives	172
Configuring the Quota System	173
Setting Quotas for Users	174
Reporting Quota Status	175
What is Software RAID?	176
Software RAID Configuration	177
Software RAID Testing and Recovery	178
What is Logical Volume Manager (LVM)?	179
Creating Logical Volumes	180
Resizing Logical Volumes	181
Logical Volume Manager Snapshots	183
Using LVM Snapshots	184
Archiving tools: tar	186
Archiving Tools: dump/restore	187
Archiving Tools: rsync :	188
End of Unit 7	189
Lab 7: Advanced Filesystem Management	190
Sequence 1: Implementing Quotas	191
Sequence 2: Software RAID	192
Sequence 3: Creating Logical Volumes with LVM	193

Sequence 4: Extending a filesystem	194
Challenge Sequence 5: Reducing a filesystem	195
Challenge Sequence 6: Backing a Logical Volume with Snapshots	196

Unit 8 - Network Configuration

Objectives	208
Network Interfaces	209
Driver Selection	211
Speed and Duplex Settings	212
IPv4 Addresses	214
Dynamic IPv4 Configuration	215
Static IPv4 Configuration	217
Device Aliases	218
Routing Table	219
Default Gateway	220
Configuring Routes	221
Verify IP Connectivity	222
Defining the Local Host Name	223
Local Resolver	224
Remote Resolvers	225
Verify DNS Connectivity	226
Network Configuration Utilities	227
Transparent Dynamic Configuration	228
Implementing IPv6	229
IPv6: Dynamic Interface Configuration	230
IPv6: Static Interface Configuration	231
IPv6: Routing Configuration	232
New and Modified Utilities	233
End of Unit 8	234
Lab 8: Manage Network Settings	235
Sequence 1: Setting up a static IP address	236
Sequence 2: Virtual IP Addresses and Static Routes	237
Sequence 3: Troubleshooting Slow DNS	238

Unit 9 - Installation

Objectives	245
Anaconda, the Red Hat Enterprise Linux Installer	246
First Stage: Starting the Installation	247
First Stage: Boot Media	248
Accessing the Installer	250
First Stage: Installation Method	251
Network Installation Server	252
Second Stage: Installation Overview	253
Configuring File Systems	254
Advanced Partitioning	255

Package Selection	256
First Boot: Post-Install Configuration	257
Kickstart	258
Starting a Kickstart Installation	259
Anatomy of a Kickstart File	260
Kickstart: Commands Section	261
Kickstart: Commands section	262
Kickstart: Packages Section	264
Kickstart: %pre, %post	265
End of Unit 9	266
Lab 9: Installation and System-Initialization	267
Sequence 1: Installing Red Hat Enterprise Linux	268
Sequence 2: Kickstart Installation	269

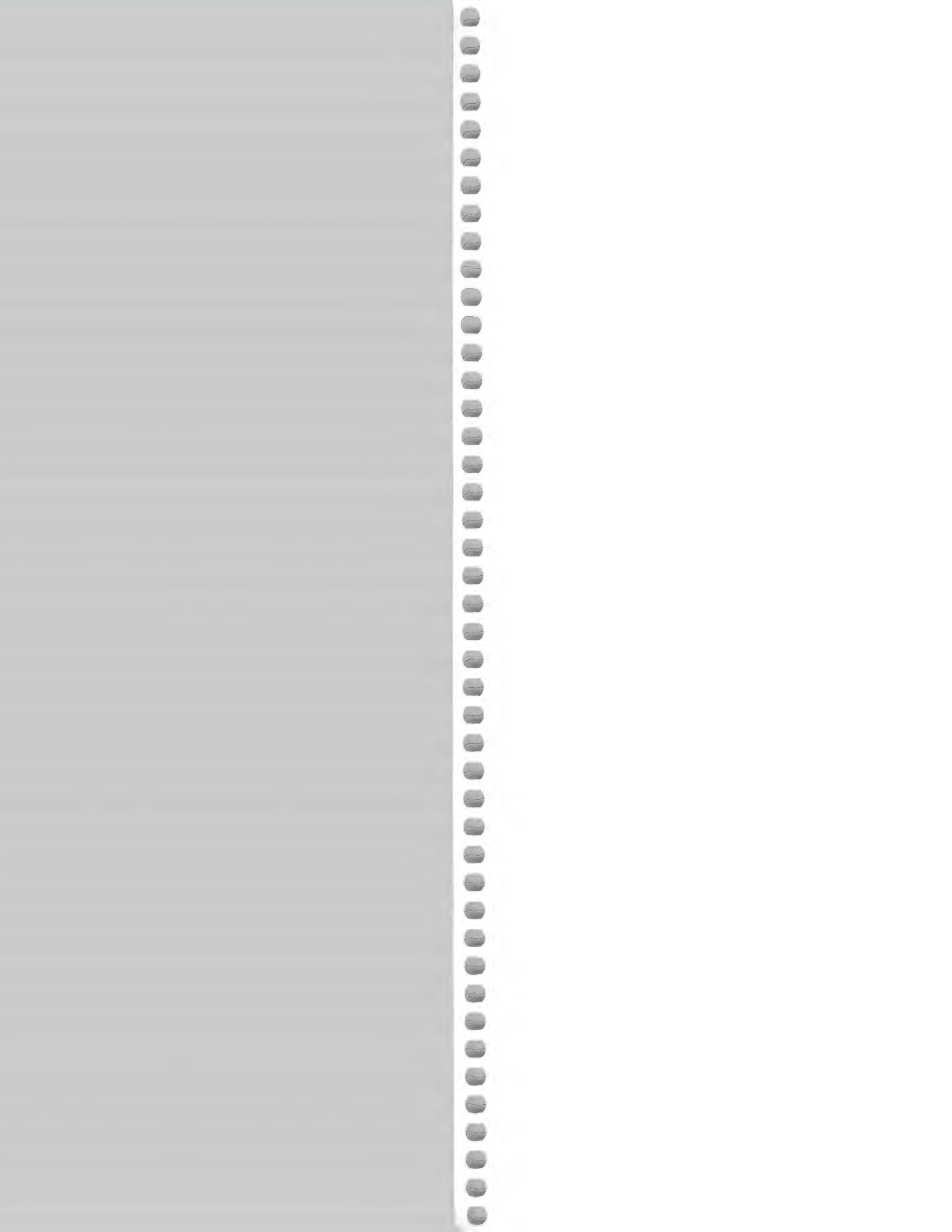
Unit 10 - Virtualization with Xen

Objectives	277
Virtualization with Xen	278
Hardware Considerations	280
Preparing Domain-0	281
Virtual Resources	282
Domain-U Configuration	283
Installing a new Domain-U	284
Domain Management with xm	285
Activating Domains on boot	286
End of Unit 10	287
Lab 10: Exploring Virtualization	288
Sequence 1: Installing the Xen Virtualization Environment	289
Sequence 2: Creating a Domain-U Virtual Machine	290
Sequence 3: Stating and Managing Domain-Us	291

Unit 11 - Troubleshooting

Objectives	296
Method of Fault Analysis	297
Fault Analysis: Gathering Data	298
Things to Check: X	299
Things to Check: Networking	300
Order of the Boot Process	301
Filesystem Corruption	302
Filesystem Recovery	303
Recovery Run-levels	304
Rescue Environment	305
Rescue Environment Utilities	306
Rescue Environment Details	307
End of Unit 11	308
Lab 11: System Rescue and Troubleshooting	309

Sequence 1: Repairing the MBR in the rescue environment	310
Sequence 2: Installing software in rescue mode	311
Sequence 3: Troubleshooting Practice	312



Introduction

RH133 Red Hat Enterprise Linux Administration



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Copyright

- The contents of this course and all its modules and related materials, including handouts to audience members, are Copyright © 2007 Red Hat, Inc.
- No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Red Hat, Inc.
- This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat, Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.
- If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed please email training@redhat.com or phone toll-free (USA) +1 866 626 2994 or +1 919 754 3700.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.





Welcome

Please let us know if you have any special needs while at our training facility.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Phone and network availability

Please only make calls during breaks. Your instructor will show you which phone to use. Network access and analog phone lines may be available; your instructor will provide information about these facilities. Please turn pagers to silent and cell phones off during class.

Restrooms

Your instructor will notify you of the location of these facilities.

Lunch and breaks

Your instructor will notify you of the areas to which you have access for lunch and for breaks

In case of Emergency

Please let us know if anything comes up that will prevent you from attending.

Access

Each facility has its own opening and closing times. Your instructor will provide you with this information.

Participant Introductions

Please introduce yourself to the rest of the class!



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Enterprise Linux

- Enterprise-targeted operating system
- Focused on mature open source technology
- 18-24 month release cycle
 - Certified with leading OEM and ISV products
- Purchased with one year Red Hat Network subscription and support contract
 - Support available for seven years after release
 - Up to 24x7 coverage plans available



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Red Hat Enterprise Linux product family is designed specifically for organizations planning to use Linux in production settings. All products in the Red Hat Enterprise Linux family are built on the same software foundation, and maintain the highest level of ABI/API compatibility across releases and errata. Extensive support services are available: a one year support contract and Update Module entitlement to Red Hat Network are included with purchase. Various Service Level Agreements are available that may provide up to 24x7 coverage with guaranteed one hour response time. Support will be available for up to seven years after a particular release.

Red Hat Enterprise Linux is released on an eighteen to twenty-four month cycle. It is based on code developed by the open source community and adds performance enhancements, intensive testing, and certification on products produced by top independent software and hardware vendors such as Dell, IBM, Fujitsu, BEA, and Oracle. Red Hat Enterprise Linux provides a high degree of standardization through its support for five processor architectures five processor architectures (Intel x86-compatible, AMD AMD64/Intel 64, Intel Itanium 2, IBM POWER, and IBM mainframe on System z).

Red Hat Enterprise Linux Variants

- Two Install Sets available
- Server Spin
 - Red Hat Enterprise Linux
 - Red Hat Enterprise Linux Advanced Platform
- Client Spin
 - Red Hat Enterprise Linux Desktop
 - Workstation Option
 - Multi-OS Option



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Currently, on the x86-compatible architecture, the product family includes:

Red Hat Enterprise Linux Advanced Platform: the most cost-effective server solution, this product includes support for the largest x86-compatible servers, unlimited virtualized guest operating systems, storage virtualization, high-availability application and guest failover clusters, and the highest levels of technical support.

Red Hat Enterprise Linux: the basic server solution, supporting servers with up to two CPU sockets and up to four virtualized guest operating systems.

Red Hat Enterprise Linux Desktop: a general purpose client solution, offering desktop applications such as the OpenOffice.org office suite and Evolution mail client. Add-on options provide support for high-end development workstations and virtualization.

Red Hat Network

- A comprehensive software delivery, system management, and monitoring framework
 - *Update Module*: Provides software updates
 - Included with all Red Hat Enterprise Linux subscriptions
 - *Management Module*: Extended capabilities for large deployments
 - *Provisioning Module*: Bare-metal installation, configuration management, and multi-state configuration rollback capabilities
 - *Monitoring Module* provides infrastructure health monitoring of networks, systems, applications, etc.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Network is a complete systems management platform. It is a framework of modules for easy software updates, systems management, and monitoring, built on open standards. There are currently four modules in Red Hat Network; the Update Module, the Management Module, the Provisioning Module, and the Monitoring Module.

The Update Module is included with all subscriptions to Red Hat Enterprise Linux. It allows for easy software updates to all your Red Hat Enterprise Linux systems.

The Management Module is an enhanced version of the Update Module, which adds additional features tailored for large organizations. These enhancements include system grouping and set management, multiple organizational administrators, and package profile comparison among others. In addition, with RHN Proxy Server or Satellite Server, local package caching and management capabilities become available.

The Provisioning Module provides mechanisms to provision and manage the configuration of Red Hat Enterprise Linux systems throughout their entire life cycle. It supports bare metal and existing state provisioning, storage and editing of kickstart files in RHN, configuration file management and deployment, multi-state rollback and snapshot-based recovery, and RPM-based application provisioning. If used with RHN Satellite Server, support is added for PXE bare-metal provisioning, an integrated network tree, and configuration management profiles.

Other Red Hat Supported Software

- Global Filesystem
- Directory Server
- Certificate Server
- Red Hat Application Stack
- JBoss Middleware Application Suite



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat offers a number of additional open source application products and operating system enhancements which may be added to the standard Red Hat Enterprise Linux operating system. As with Red Hat Enterprise Linux, Red Hat provides a range of maintenance and support services for these add-on products. Installation media and software updates are provided through the same Red Hat Network interface used to manage Red Hat Enterprise Linux systems.

Global Filesystem: an open source cluster file system appropriate for enterprise deployments, allowing servers to share a common pool of storage.

Directory Server: an LDAP-based server that centralizes directory storage and data distribution, such as user and group data.

Certificate Server: identity management software, using the Red Hat Directory Server as its back-end LDAP data repository.

Red Hat Application Stack: the first fully integrated open source stack, supplying everything needed to run standards based web applications, including Red Hat Enterprise Linux, JBoss Application Server with Tomcat, JBoss Hibernate, and a choice of open source databases: MySQL or PostgreSQL, and Apache Web Server.

JBoss Middleware Application Suite: a suite of applications that provide a complete middleware solution.

For additional information, see the following web pages:

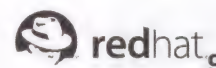
- Global Filesystem: <https://www.redhat.com/solutions/gfs/>
- Directory Server: <https://www.redhat.com/solutions/directoryserver/>
- Red Hat Application Stack: <https://www.redhat.com/solutions/rhappstack/>
- JBoss Middleware Application Suite: <https://www.redhat.com/jboss/>

Notes on Internationalization

- Red Hat Enterprise Linux supports nineteen languages
- Default language can be selected:
 - During installation
 - With **system-config-language**
 - System->Administration->Language
- Alternate languages can be used on a per-command basis:

```
$ LANG=en_US.UTF8 date
```

- Language settings are stored in `/etc/sysconfig/i18n`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Enterprise Linux 5 supports nineteen languages: English, Bengali, Chinese (Simplified), Chinese (Traditional), French, German, Gujarati, Hindi, Italian, Japanese, Korean, Malayalam, Marathi, Oriya, Portuguese (Brazilian), Punjabi, Russian, Spanish and Tamil. Support for Assamese, Kannada, Sinhalese and Telugu are provided as technology previews.

A system's language can be selected during installation, but the default is US English. To use other languages, you may need to install extra packages to provide the appropriate fonts, translations and so forth. These can be selected during system installation or with **system-config-packages** (Applications->Add/Remove Software).

The currently selected language is set with the `LANG` shell variable. Programs read this variable to determine what language to use for output:

```
[student@stationX ~]$ echo $LANG
ru_RU.UTF8
```

A system's default language can be changed with **system-config-language** (System+Administration->Language), which affects the `/etc/sysconfig/i18n` file.

Languages with non-ASCII characters may have problems displaying in some environments. Kanji characters, for example, may not display as expected on a virtual console. Individual commands can be made to use another language by setting `LANG` on the command-line:

```
[student@stationX ~]$ LANG=en_US.UTF8 date
Thu Feb 22 13:54:34 EST 2007
```

Subsequent commands will revert to using the system's default language for output.

SCIM (Smart Common Input Method) can be used to input text in various languages under X if the appropriate language support packages are installed. Type *Ctrl-Space* to switch input methods.

The Fedora Project

- Red Hat sponsored open source project
- Focused on latest open source technology
 - Rapid four to six month release cycle
 - Available as free download from the Internet
- An open, community-supported proving ground for technologies which may be used in upcoming enterprise products
- Red Hat does not provide formal support



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

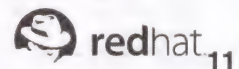
The Fedora Project is a community supported open source project sponsored by Red Hat intended to provide a rapidly evolving, technology-driven Linux Distribution with an open, highly scalable development and distribution model. It is designed to be an incubator and test bed for new technologies that may be used in later Red Hat enterprise products. The basic Fedora Core distribution will be available for free download from the Internet.

The Fedora Project will produce releases on a short four to six month release cycle, to bring the latest innovations of open source technology to the community. This may make it attractive for power users and developers who want access to cutting-edge technology and can handle the risks of adopting rapidly changing new technology. Red Hat does not provide formal support for the Fedora Project.

For more information, visit <http://www.fedoraproject.org>.

Classroom Network

	Names	IP Addresses
Our Network	example.com	192.168.0.0/24
Our Server	server1.example.com	192.168.0.254
Our Stations	stationX.example.com	192.168.0.X
Hostile Network	cracker.org	192.168.1.0/24
Hostile Server	server1.cracker.org	192.168.1.254
Hostile Stations	stationX.cracker.org	192.168.1.X
Trusted Station	trusted.cracker.org	192.168.1.21



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `server1` system provides a number of services to the classroom network, including:

- A DHCP server
- A web server. The web server distributes RPMs at `http://server1.example.com/pub`.
- An FTP server. The FTP server distributes RPMs at `ftp://server1.example.com/pub`.
- An NFS server. The NFS server distributes RPMs at `nfs://server1.example.com/var/ftp/pub`.
- An NTP (network time protocol) server, which can be used to assist in keeping the clocks of classroom computers synchronized.
- A sendmail server, which can be used to store e-mail inboxes. The sendmail server does not provide this service to the classroom by default; the instructor will configure it if it is needed in the classroom.
- An NIS (network information systems) server, which can be used to distribute account information. The NIS server is turned off by default; the instructor will enable it if it is needed in the classroom.

Objectives

- Understand system and service initialization
- Integrate new filesystems
- Understand advanced partitioning schemes
- Perform filesystem management tasks
- Set up networking
- Perform user and group administration
- Automate tasks with at, cron, and anacron
- Set up core services: Logging, Printing, X Window system
- Manage software packages with **yum** and **rpm**
- Install the system interactively and with Kickstart
- Perform basic troubleshooting



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Audience and Prerequisites

- Audience: Linux or UNIX users, who understand the basics of Red Hat Linux, that desire further technical training to begin the process of becoming a system administrator.
- Prerequisites: RH033 Red Hat Linux Essentials or equivalent experience with Red Hat Enterprise Linux.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



Unit 1

System Initialization



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Objectives

Upon completion of this unit, you should be able to:

- Discuss the boot sequence
- Understand GRUB's role
- Understand init's role
- Control System V services



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Boot Sequence Overview

- BIOS Initialization *pdf*
- Boot Loader
- Kernel initialization
- **init** starts and enters desired run level by executing:
 - `/etc/rc.d/rc.sysinit`
 - `/etc/rc.d/rc` and `/etc/rc.d/rc?.d/`
 - `/etc/rc.d/rc.local`
 - X Display Manager if appropriate



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Each major step in a Linux system's boot sequence - BIOS initialization, Boot loader, kernel initialization, and init startup - is covered in the upcoming pages.

Boot Loader Components

- Boot Loader
 - 1st Stage - small, resides in MBR or boot sector
 - 2nd Stage - loaded from boot partition
- Minimum specifications for Linux:
 - Label, kernel location, OS root filesystem and location of the initial ramdisk (*initrd*)
- Minimum specification for other OS:
 - boot device, label



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The boot loader is responsible for loading and starting your Linux operating system (or possibly other operating systems) when the computer is started up.

The boot loader is generally invoked in one of two ways:

- BIOS passes control to an initial program loader (IPL) installed within a drive's Master Boot Record.
- BIOS passes control to another boot loader, which passes control to an IPL installed within a partition's boot sector.

In either case, the IPL (initial program loader) must exist within a very small space, no larger than 446 bytes. Therefore, the IPL for GRUB is merely a first stage, whose sole task is to locate and load a second stage boot loader, which does most of the work to boot the system.

There are two possible ways to configure boot loaders:

- primary boot loader: Install the first stage of your Linux boot loader into the Master Boot Record. The bootloader must be configured to pass control to any other desired operating systems.
- secondary boot loader: Install the first stage of your Linux boot loader into the boot sector of some partition. Another bootloader must be installed into the MBR, and configured to pass control to your Linux boot loader.

GRUB and grub.conf

- GRUB “the GRand Unified Bootloader”
 - Command-line interface available at boot prompt
 - Boot from ext2/ext3, ReiserFS, JFS, FAT, minix, or FFS file systems
 - Supports MD5 password protection
- /boot/grub/grub.conf
- Changes to grub.conf take effect immediately
- If MBR on /dev/hda is corrupted, reinstall the first stage bootloader with:
 - /sbin/grub-install /dev/hda

1st IDE HD



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

/boot/grub/grub.conf has a format of global options followed by boot stanzas. Here is a sample grub.conf:

```
timeout=5
splashimage=(hd0,0)/grub/splash.xpm
hiddenmenu
password --md5 $1$/iX9y$Bk4yt37Ch2fZ5GFA
default=0

title Red Hat Enterprise Linux AS (2.6.9-648_EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-648.EL ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    initrd /initrd-2.6.9-648.EL.img

title Windows XP Pro
    rootnoverify (hd0,1)
    chainloader +1
```

Changes to grub.conf take effect immediately. GRUB reads the configuration file at boot time, so the grub.conf file must be stored on a filesystem GRUB understands. These include ext2/ext3, reiserfs, FAT, minix, and FFS.

The MD5 password hash can be created with **grub-md5-crypt**.

If for some reason your MBR becomes corrupted and you need to reinstall GRUB, you can do so with the command **/sbin/grub-install boot-device**. Occasionally it may prove necessary for the user to set up grub manually. If **grub-install** fails for some reason try the following:

1. Type the command **grub** and press *Enter*
2. Type **root (hd0,0)**
3. Type **setup (hd0)**
4. Type **quit**

Starting the Boot Process: GRUB

- Image selection
 - Select with space followed by up/down arrows on the boot splash screen
- Argument passing
 - Change an existing stanza in menu editing mode
 - Issue boot commands interactively on the GRUB command line



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The GRUB Boot Screen

When GRUB starts up, a graphical splash screen can be accessed by pressing *Return*; or *Space*. This screen has a list of menu entries, normally bootable images. You can select between the different images with the up and down arrow keys, and press *Return* to select a particular entry for booting.

If you want to pass arguments to boot images through menu editing mode or access the GRUB command line, and a GRUB password is set, you will need to type *p* followed by your GRUB password.

Menu Editing Mode

If you then select an entry and type *e*, you will be dropped into menu editing mode. This mode allows you to modify an existing boot stanza to pass options to the kernel or init, or select alternate root file systems or kernel files than you have configured in your existing stanzas. You can use arrow keys to select a line, *e* to edit a line, *d* to delete a line, *o* to add a line, and *b* to boot. For example, to boot into runlevel 2, you could select menu editing mode, select your Linux boot stanza, add a 2 to the end of your kernel line, and type *b* to boot the modified menu entry.

The GRUB Command Line

GRUB provides a command-line interface which can be used to write a temporary boot command from scratch, view the contents of files on the filesystem, perform diagnostic tests, or experiment with GRUB configurations. Most commands supported by the configuration file are available for interactive use. Editing commands are similar to those used by the bash shell, and *Tab* completion is available. If GRUB is not able to find a valid `grub.conf` file, it will default to the command line.

To exit menu editing mode or the command line and go back to the main GRUB menu, type *Esc*.

For more information about GRUB and `grub.conf`, look at **info grub**.

Handwritten notes:
title = stanza
kernel stanza of sec
hidden menu
kernel (hd0,0)
e2label
partitions

Kernel Initialization

- Kernel boot time functions
 - Device detection
 - Device driver initialization
 - Mounts root filesystem read only
 - Loads initial process (init)



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The kernel initialization files generate good output, but scroll by quickly. A good way to examine this output is to view `/var/log/dmesg`, which contains a snapshot of these kernel messages taken just after control is passed to `init`. Review of this output will reveal the basic initialization steps of the Linux kernel:

Device drivers compiled into the kernel are called, and will attempt to locate their corresponding devices. If successful in locating the device, the driver will initialize and usually log output to the kernel message buffer.

If essential (needed for boot) drivers have been compiled as modules instead of into the kernel, then they must be included in an `initrd` image, which is then temporarily mounted by the kernel on a RAM disk to make the modules available for the initialization process.

After all the essential drivers are loaded, the kernel will mount the root filesystem read-only.

The first process is then loaded (`init`) and control is passed from the kernel to that process.

init Initialization

- init reads its config: `/etc/inittab`
 - initial run level
 - system initialization scripts
 - run level specific script directories
 - trap certain key sequences
 - define UPS power fail / restore scripts
 - spawn gettys on virtual consoles
 - initialize X in run level 5



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

init

init is the parent of all processes. This is easily shown by running the **ps**tree command:

```
[student@stationX]$ ps tree
init--apmd
|  -atd
|  -automount
|  -crond---crond
|  -deskguide_apple
|  -gdm--X
|      ^-gdm---gnome-session
```

Because init is the first process, it will always have a PID of number 1.

The file `/etc/inittab` contains the information on how init should set up the system in every run level, as well as the run level to use as default.

If the `/etc/inittab` file is missing or seriously corrupt, you will not be able to boot to any of the standard run levels (0-6) and will need to use single or emergency mode instead. This procedure is discussed in depth in the Troubleshooting unit of this course.

Run Levels

- init defines run levels 0-6, S, emergency
- The run level is selected by either
 - the default in `/etc/inittab` at boot
 - passing an argument from the boot loader
 - using the command `init new_runlevel`
- Show current and previous run levels
 - `/sbin/runlevel`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The following chart details the run levels that Linux defines by default:

0	halt
1	single user mode
2	multiuser mode without NFS
3	full multiuser mode, typical for servers
4	officially undefined
5	graphical login, typical for desktops and laptops
6	reboot
s, S, single	alternate single user mode
emergency	bypass rc.sysinit , su login

K. M. S. C. P. S.
Shri Ram

`/etc/rc.d/rc.sysinit`

- Important tasks include:
 - Activate `udev` and `selinux`
 - Sets kernel parameters in `/etc/sysctl.conf`
 - Sets the system clock
 - Loads keymaps
 - Enables swap partitions
 - Sets hostname
 - Root filesystem check and remount
 - Activate RAID and LVM devices
 - Enable disk quotas
 - Check and mount other filesystems
 - Cleans up stale locks and PID files



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

When `rc.sysinit` first starts, it prompts you to press the letter `i` if you want to enter interactive mode. In interactive mode, it will prompt you for confirmation before performing most of its functions. The script is straightforward in layout, and well documented, so for additional details simply look at the script source directly.

`/etc/rc.d/rc`

- Initializes the default run level per the `/etc/inittab` file `initdefault` line such as `id:3:initdefault:`
- ```
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3 (default)
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `rc` script initializes the intended run level as defined by the `initdefault` line in the `/etc/inittab` and it is responsible for starting/stopping services when the runlevel changes.

*So it's responsible for  
distributed to the*

# System V run levels

- Run level defines which services to start
  - Each run level has a corresponding directory:
    - `/etc/rc.d/rcX.d`
  - The System V init scripts reside in:
    - `/etc/rc.d/init.d`
  - Symbolic links in the run level directories call the `init.d` scripts with a start or stop argument



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The ability to change run levels allows easy interaction for administrators. If you need to test some software or perform system maintenance you can switch to run level 1 and the system will be in single-user mode. You do not have to shutdown and restart to change run levels. The scripts that launch the system services are located in the `/etc/rc.d/init.d`.

```
$ ls -l /etc/rc.d/init.d
```

```
-rwxr-xr-x 1 root root 1535 Jul 15 12:09 amd
-rwxr-xr-x 1 root root 798 Aug 4 03:01 anacron
-rwxr-xr-x 1 root root 1289 Aug 17 03:15 apmd
-rwxr-xr-x 1 root root 908 Aug 11 15:43 arpwatc
... output truncated ...
```

```
$ ls -l /etc/rc.d/rc3.d
```

```
lrwxrwxrwx 1 root root 14 Sep 22 16:15 K15pvmd -> ../init.d/pvmd
lrwxrwxrwx 1 root root 13 Sep 22 16:11 K20nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root 15 Sep 22 16:09 S05kudzu -> ../init.d/kudzu
lrwxrwxrwx 1 root root 18 Sep 22 16:08 S08ipchains -> ../init.d/ipchains
lrwxrwxrwx 1 root root 17 Sep 22 16:04 S10network -> ../init.d/network
lrwxrwxrwx 1 root root 16 Sep 22 16:04 S12syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 17 Sep 22 16:11 S13portmap -> ../init.d/portmap
... output truncated ...
```



## `/etc/rc.d/rc.local`

- Run after the run level specific scripts
- Common place for custom modification
- In most cases it is recommended that you create a System V *init* script in
- `/etc/rc.d/init.d` unless the service you are starting is so trivial it doesn't warrant it. Existing scripts can be used as a starting point.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

### **rc.local - Final System V Initialization**

Because the **rc.local** script is run each time the system enters a run level, it is a convenient place to start processes that need to be running.

# Controlling Services

- Utilities to control default service startup
  - **system-config-services**: graphical utility that requires an X interface
  - **ntsysv**: ncurses based utility usable in virtual consoles
  - **chkconfig**: a fast, versatile command line utility that works well and is usable with scripts and Kickstart installations
- Utilities to control services manually
  - **service**: immediately start or stop a standalone service
  - **chkconfig** immediately starts and stops xinetd-managed services



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The job of the System V initialization scripts is to start services at boot time. Most of these services run as daemons, such as cups, cron and sendmail. Red Hat Enterprise Linux includes several utilities that facilitate the management of System V initialization.

**system-config-services** is an X client that presents a display of each of the services that are started and stopped at each run level. Services can be added, deleted, or re-ordered in run levels 3 through 5 with this utility.

**ntsysv** is a console-based interactive utility that allows you to control what services run when entering a given run level. This utility is used during system installation, but can be run from the command line. It configures the current run level by default. By using the `--level` option you can configure other run levels.

**chkconfig** is a command-line utility. When passed the `--list` switch, it displays a list of all System V scripts and whether each one is turned on or off at each run level. Scripts can be managed at each run level with the `on` and `off` **chkconfig** directives. The `--level` option can be used to specify the runlevels affected if the defaults are unacceptable.

The **service** command is used to start or stop a standalone service immediately; most services accept the arguments `start`, `stop`, `restart`, `reload`, `condrestart`, and `status` as a minimum.

The **system-config-services** and **chkconfig** commands will start or stop an xinetd-managed service as soon as you configure it on or off. Standalone services will not start or stop until the system is rebooted or you use the **service** command.

# End of Unit 1

- Questions and Answers
- Summary
  - System BIOS
  - GRUB
  - init
  - chkconfig and service



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.







# Lab 1

## Managing Startup

---

Goal: To familiarize yourself with the startup process

System Setup: A system installed with Red Hat Enterprise Linux



## Sequence 1: Changing the default run level

**Deliverable:** A system that boots to runlevel 3 by default.

**Instructions:**

1. Change the default runlevel to level 3 and reboot.
2. Change the default run level back to level 5 and reboot.

## Sequence 1 Solutions

1. Change the default runlevel to level 3 and reboot.
  - a. Edit the `/etc/inittab` file and change the default run level as shown below from level 5 to level 3.

*- no signal IK*  
`id:3:initdefault:`

- b. Reboot the system.

2. Change the default run level back to level 5 and reboot.

`id:5:initdefault:`









## Unit 2

# Package Management



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



# Objectives

Upon completion of this unit, you should be able to:

- Install and remove RPM packages
- Query packages and verify their state
- Manage packages using **yum**
- Understand the relationship between **yum** and **rpm**
- Configure **yum** to connect to a RHN Satellite Server
- Create a private **yum** repository
- Configure **yum** to connect to a private repository
- Configure and use Red Hat Network



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# RPM Package Manager

- RPM Components
  - local database
  - **rpm** and related executables
  - RPM frontends such as yum
  - package files
- Primary Functions
  - install/remove
  - query
  - verify



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The RPM Package Manager greatly simplifies the distribution, installation, upgrading, and removal of software on Red Hat Enterprise Linux systems. The RPM system consists of a local database, the rpm executable, rpm package files.

The local RPM database is maintained in `/var/lib/rpm`. The database stores information about installed packages such as file attributes and package prerequisites. An administrator rarely, if ever, modifies the database directly, but instead uses the rpm command.

Software to be installed using **rpm** is distributed through rpm package files, which are essentially compressed archives of files and associated dependency information. Package files are named using the following format:

`name-version-release.architecture.rpm`

The `version` refers to the open source version of the project, while the `release` refers to Red Hat internal patches to the open source code.

In contrast to package management on some other platforms, RPM's design does not provide interactive configuration of software as part of the package load process. RPM can perform configuration actions as part of the installation, but these are scripted, not interactive. It is common for packages to install with reasonable default configurations applying. On the other hand, some software installs in an unconfigured state.

**rpm** is a backend for other programs such as **yum** or **system-config-packages**. These tools provide significant advantages such as automatic dependency resolution.



# Installing and Removing Software

- Primary RPM options:
  - Install: **rpm -i, --install**
  - Upgrade: **rpm -U, --upgrade**
  - Freshen: **rpm -F, --freshen**
  - Erase: **rpm -e, --erase**
- Output Options: **-v, -h**
- URL support: **ftp://** (with globbing), **http://**
- Many other install-options are available to address special cases.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Installing: rpm -i

The primary function of RPM is to install, upgrade, and remove software from a system. A package is installed using a command such as **rpm -i#zip-2.3-8.i386.rpm**

When installing an rpm package, the rpm command will consult the local database to ensure that (1) any prerequisites (in the form of files, libraries, rpms, or generally defined "provisions") are installed on the system, and (2) installing the rpm will not clobber any preexisting files. The checks can be omitted by enabling the **--nodeps** or **--replacefiles** command-line switches, respectively, or both using the **--force** switch. rpm will provide "pretty" output if called with the **-v** (print package name) and **-h** (print hash marks) options.

## Upgrading: rpm -U and rpm -F

rpm can be used to upgrade already installed software with the **-U (--upgrade)** command-line switch. When upgrading, the original package (with the exception of configuration files) on the system will be removed, and the new package installed. Configuration files from the original installation are saved with a **.rpm\_save** extension.

Freshening is almost identical to upgrading, except when the package specified on the command line is not already installed on the system. When upgrading with **-U**, the package will be installed whether or not it is already installed; when freshening, the package will be ignored if not already installed. To apply all errata released by Red Hat for all packages installed on your system, ignoring errata for uninstalled packages, execute the following:

## Uninstalling: rpm -e

Software is removed from your system using the **-e (--erase)** command-line switch. The package argument must be the installed package's name, not the package file name. For example, these commands first install the zip package file, and then remove it:

```
rpm -ihv zip-2.3-8.i386.rpm
```

```
rpm -e zip
```



# Updating a Kernel RPM

- Make sure to install kernel updates
- Do not use **rpm -U** or **rpm -F** !
  - **rpm -ivh kernel-version.arch.rpm**
  - Boot new kernel to test
  - Revert to old kernel if a problem arises
  - **rpm -e kernel-oldversion** if no problems



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Installing a new kernel is one of the few things you will do on your system that requires a reboot operating systems. It also requires a little more thought and caution, as it is quite simple to render a system temporarily inoperable if you are careless when updating the kernel. Unlike just about any other upgrade you might do, you should NOT upgrade the kernel using **rpm -U** or **-F**.

Recall how **rpm** functions with the **-U** and **-F** options: it determines whether a version already exists on the system, and if so, whether the version to be installed is newer. If it is newer, **rpm** first installs the new version, thereby replacing old files -- except those marked as configuration files. It then removes the old package, but only deletes files that do not exist in the new version.

Because upgrading removes the previous kernel version, if your newly-installed kernel proves unstable, you could be left with an unbootable system. You would have to resort to alternate boot media such as a boot floppy or the CD-ROM. When you run an install instead of an upgrade, the old version of the kernel is still available and can be selected from the boot loader.

In addition, kernel modules are version-specific, and an upgrade will remove all modules that your present kernel is using, leaving the system unable to dynamically load device drivers or other modules.

Because all of the kernel RPM's files are version-specific (that is, they either include version information in their names or are stored in version-specific paths), it is possible to install multiple versions of the kernel package. If you use **rpm -ivh**, instead of **-U**, then the new kernel will be added to your system, but your old kernel remains on it as well.

By default, the new kernel is automatically added to GRUB. You can change this behavior by editing `/etc/sysconfig/kernel`.

# rpm Queries

- Syntax:
  - **rpm -q what\_packages what\_information**
- Installed Package Options:
  - **rpm -qa** lists installed packages
  - **rpm -qf filename#**shows owning package
  - **rpm -qi package\_name#**general information
  - **rpm -ql package\_name#**lists files in package
- Uninstalled Package Options:
  - **rpm -qip package\_file.i386.rpm**
  - **rpm -qlp package\_file.i686.rpm**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

RPM provides robust querying, which is invoked with **rpm -q** or **rpmquery**. Query options fall into one of two categories: those that specify which packages to query, and those that specify what information to retrieve. The first must be specified; the second defaults to the package name.

Here's a list of common package specification parameters and what they return:

|                                  |                                         |
|----------------------------------|-----------------------------------------|
| <b>-qa</b>                       | all installed packages                  |
| <b>-q package_name</b>           | the named package and version           |
| <b>-qf filename</b>              | the package that owns the file          |
| <b>-qp package_file_name</b>     | the (possibly uninstalled) package file |
| <b>--whatrequires capability</b> | all packages that require capability    |
| <b>--whatprovides capability</b> | all packages that provide capability    |

The following is a list of common query information parameters and what they return:

|                      |                                                                                                                  |
|----------------------|------------------------------------------------------------------------------------------------------------------|
| <b>-i</b>            | general package information                                                                                      |
| <b>-l</b>            | package files                                                                                                    |
| <b>--requires</b>    | package prerequisites                                                                                            |
| <b>--provides</b>    | capabilities provided by package                                                                                 |
| <b>--scripts</b>     | scripts run upon installation and removal                                                                        |
| <b>--changelog</b>   | package revision history                                                                                         |
| <b>--queryformat</b> | format custom-formatted information (use <b>rpm --querytags</b> to list available tags for use in format string) |



# rpm Verification

- Installed RPM File Verification:
  - **rpm -V <package\_name>**
  - **rpm -Vp <package\_file>.i386.rpm**
  - **rpm -Va**
- Signature verification BEFORE package install:
  - **rpm --import RPM-GPG-KEY**
  - **rpm -K <package\_file>.i386.rpm**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## RPM file verification

Verifying an installed package compares the file sizes, permissions, type, owner, group, MD5 checksum, and modify time against the RPM database. Any inconsistencies will be reported. An installed package can also be verified against a package file as well:

**rpm -V zip** - verifies the installed zip rpm against the RPM database

**rpm -Va** - verifies all installed RPMS against the RPM database

**rpm -Vp zip-2.3-8.i386.rpm** - verifies the installed zip package against the zip package file

## RPM signature verification

Red Hat signs all package files with a GPG private signature. The complementary public signature is shipped with every Red Hat distribution. To verify the integrity of any package file, you must first import the Red Hat public key. The rpm utility will automatically verify the signature of any package you install at install time. You can also check the integrity of package files using the **--checksig** option.

**rpm --import /mnt/cdrom/RPM-GPG-KEY**

**rpm -qa gpg-pubkey**



# About yum

- Front-end to **rpm**
  - Designed to resolve package dependencies
  - Can locate packages across multiple repositories
- Replacement for **up2date**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Development of RPM cemented the future of Linux by greatly simplifying installation of software. As the operating system became more complex, RPM began to show a few weaknesses, primarily its inability to resolve dependencies.

```
[root@stationX ~]# rpm -ivh x3270-x11-*
warning: x3270-x11-3.3.4p7-3.el5.1.x86_64.rpm: Header V3 DSA signature: NOKEY, key ID 897da07a
error: Failed dependencies:
 x3270 = 3.3.4p7 is needed by x3270-x11-3.3.4p7-3.el5.1.x86_64
[root@stationX Server]#
```

Unfortunately, there is no way to look at the above output and determine if the suggested RPM, x3270, also has a dependency. As a matter of fact it is possible that dozens of RPMs would have to be installed. If the need RPMs were not available in the current directory, it would be up to the user to locate and install each.

To solve the problem of dependency resolution and package location, volunteer programmers at Duke University developed Yellow dog Update, Modified, or YUM for short. The system is based on repositories that hold RPMs and a repodata filelist. The yum application can call upon several repositories for dependency resolution, fetch the RPMs, and install the needed packages.

```
[root@stationX ~]# yum install x3270-x11
... output truncated ...
Dependencies Resolved
```

```
=====
Package Arch Version . Repository Size
=====
Installing:
 x3270-x11 x86_64 3.3.4p7-3.el5.1 server1 424 k
Installing for dependencies:
 x3270 x86_64 3.3.4p7-3.el5.1 server1 142 k
```

## Transaction Summary

```
=====
Install 2 Package(s)
Update 0 Package(s)
Remove 0 Package(s)
```

```
Total download size: 566 k
Is this ok [y/N]:
```

# Using yum

- Install/Remove/Update
  - **yum install** *package...*
  - **yum remove** *package...*
  - **yum update** [*package...*]



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The command-line utility **yum** gives you an easy way to manage the packages on your system:

```
[root@stationX ~]# yum install firefox
```

The above command will search the configured repositories for a package named **firefox**, and if found will install the latest version, pulling in dependencies if needed.

```
[root@stationX ~]# yum remove mypackage
```

The above command will try to remove the package named **mypackage** from your system. If any other package depends on **mypackage** **yum** will prompt you about this, giving you the option to remove those packages as well.

```
[root@stationX ~]# yum update [mypackage...]
```

If any packages are specified on the command-line **yum** will search the configured repositories for updated versions of those packages and install them. When no packages are specified **yum** will search for updates to all of your currently installed packages.



# Searching packages/files

- Searching packages
  - **yum search** *searchterm*
  - **yum list** (*all/available/extras/installed/recent/updates*)
  - **yum info** *packagename*
- Searching files
  - **yum whatprovides** *filename*



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**yum search** *searchterm* will search all known package names and descriptions for *searchterm*:

```
[root@stationX ~]# yum search cairo
```

**yum list** *searchterm* will search all known package names for *searchterm*. *searchterm* can include wildcards:

```
[root@stationX ~]# yum list '*irefo*'
```

**yum info** *package...* will search all the package database for *package* and display some info about the package.

```
[root@stationX ~]# yum info '*irefo*'
```

**yum whatprovides** *filename* will search all packages (both installed and available) for *filename*. This can be useful when you know the filename of an executable/library you need, but you don't know the package name.

```
[root@stationX ~]# yum whatprovides /usr/sbin/sendmail
```



# Configuring Additional Repositories

- Create a file in `/etc/yum.repos.d` for your repository
- Required information
  - `[repo-name]`
  - `name=A nice description`
  - `baseurl=http://yourserver.com/path/to/repo`
  - `enabled=1`
  - `gpgcheck=1`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Inside a repository declaration you can use variables like `$releasever` and `$basearch` to be substituted with the relevant information for your installation. Using this you can roll out one repository-file for use on multiple architectures or releases.

## Creating a private repository

- Create a directory to hold your packages
- Make this directory available by http/ftp
- Install the **createrepo** RPM
- Run **createrepo -v /package/directory**
- This will create a `repodata` subdirectory and the needed support files



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The **createrepo** creates the support files necessary for a yum repository. These files will put into the `repodata` subdirectory.

`repomd.xml` - The `repomd.xml` contains timestamps and checksum values for the other three files. Once a client has established a connection with a server, it caches all files, and only refreshes the cache if `repomd.xml` indicates the repo has changed.

`primary.xml.gz` - The `primary.xml.gz` file contains the list of all the RPMs in the repository, as well as dependency information. It also contains the information that would normally be returned by **rpm -qlp**.

`filelists.xml.gz` - The `filelists.xml.gz` file contains a list of all the files in all the RPMS. This is used by queries such as **yum whatprovides**.

`other.xml.gz` - The `other.xml.gz` file contains additional information, including the changelogs for the RPMs.

`comps.xml` - The optional `comps.xml` file contains information about package groups. This allows group installations and optimizes dependency resolution.

The addition or deletion of files within the repository requires a **createrepo** to be run again.

# Red Hat Network

- Centralized platform for systems management
  - provides Red Hat software packages
  - shows if errata are available for systems
  - can update many systems at once
  - allows full life cycle management
- Webbased management interface
- Uses HTTPS for all transactions



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Red Hat Network allows administrators to manage software installation and upgrades efficiently using a combination of your RHN account and the **up2date** utility.

In March of 2001, the Lion worm, a self-spreading program intended to compromise security on Linux systems, made headlines. According to a report found at <http://www.sans.org>, the worm makes use of a “BIND vulnerability... that was reported back on January 29th, 2001”. Red Hat had posted an updated version of the bind RPM that same day in January. If all Linux administrators had promptly updated their systems, the Lion worm would not have made headlines. Red Hat Network attempts to simplify the task of keeping software updated, reducing a system's vulnerability to exploits.



# Red Hat Network Server

- `rhn.redhat.com` or local Satellite/Proxy
  - Web based management of machines
  - RHN Proxy caches RHN traffic
  - RHN Satellite provides an autonomous RHN
- RHN Accounts
  - RHN Users for registration of machines and web based management
  - System ID for automatic authentication of systems



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

RHN Service is provided either by `rhn.redhat.com` or a local Satellite or Proxy server. The server provides the packages and offers remote management of the clients. Transactions (i.e. installation of a package) can be scheduled on the web interface, which are executed by clients the next time they poll RHN.

## RHN Proxy Server

Software updates and errata may be locally cached using a RHN Proxy server. Client profiles are still maintained on RHN servers. The base channel (i.e., “Red Hat Enterprise Linux (v. 5 for 32-bit x86)”) is managed by RHN, but additional private sub-channels that allow the local distribution of custom software can be defined and locally administered. All interactions with RHN are mediated by the RHN Proxy Server, so only the proxy server needs Internet access.

## RHN Satellite Server

The RHN Satellite Server allows all aspects to be managed locally, including client profiles and custom channel management. Systems can be provisioned from bare metal with the Provisioning Entitlement. System management is performed using a local web server, and a local database maintains client accounts and profiles. The Satellite Server allows complete channel definition, control, and management. No Internet access is required.

## Accounts

Users login to the RHN web interface to manage the systems online. The RHN User is also used for registration of new systems. Upon registration a system-id is generated for the system, so that the system can connect to RHN noninteractively.

# Entitlements

- Grant access to software channels
  - Base Channel
  - Child Channel(s)
- Define level of service
  - Update
  - Management
  - Provisioning
  - Monitoring



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Channels

Entitlements define which software channels can be subscribed by a system. There are two different types of software channels. Systems are assigned to one base channel like Red Hat Enterprise Linux (v.5 for 32bit x86).

Child channels may be subscribed in addition. The Fast Track Child Channel, which is available to all systems provides bugfix errata before they are integrated into a maintenance release.

## Level of Service

Update Entitlements are included with all Red Hat Enterprise Linux subscriptions. They allow management of machines by a single administrator

Management Entitlements allow for the grouping of client systems, including collective software management and errata notifications. Multiple administrators may be defined and assigned to the various system groups.

With Provisioning Entitlements configuration files and Kickstart profiles can be managed.

Systems with Monitoring Entitlements can be monitored by RHN Satellite. Monitoring includes system activity, availability. If defined



# Red Hat Network Client

- Registration
  - Run **rhnc\_register**
  - Select the updates location (RHN or local satellite/proxy)
  - Enter Account information
- Interactive usage
  - **yum** plugin for downloading packages from RHN
  - Configuration in `/etc/yum/pluginconf.d/rhn-plugin.conf`
- Remote management
  - **rhnsd** polls RHN every four hours
  - **rhnc\_check** polls immediately



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**rhnc\_register** will create configuration files for **yum** as well as registering your system with RHN or your satellite-server.

RHN can be used to perform remote administration of collections of machines. First, actions for the machine (such as specific package installation or upgrades) are queued for the machine using the RHN account.

Client machines use the **rhnsd** daemon to poll RHN periodically for queued actions. By default, **rhnsd** polls every 4 hours, though this can be adjusted in `/etc/sysconfig/rhn/rhnsd`. The **rhnsd** daemon uses the `/usr/sbin/rhnc_check` command to actually perform the poll and administer any queued actions. Notably, the **rhnsd** daemon does not open any server networking ports.



## End of Unit 2

- Questions and Answers
- Summary
  - What are the primary functions of RPM?
  - What **rpm** options should be used to install a kernel RPM?
  - Package-management with yum
  - Relationship between **yum** and **rpm**
  - Using **yum** with RHN
  - Creating a private repository
  - Configuring repositories
  - How does Red Hat Network work?



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.









# Lab 2

## Working with packages

---

|               |                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------|
| Goal:         | To gain working experience with package management                                                                      |
| System Setup: | A working install of Red Hat Enterprise Linux 5 connected to the classroom network                                      |
| Situation:    | You have been asked to connect a system to your company's private <b>yum</b> repository to install and update software. |

## Sequence 1: Using RPM

### Instructions:

1. Use **rpm** queries to answer the following questions. In the blank spaces, write in the command used to find the answers.

What files are in the initscripts package?

rpm -ql initscripts

On what host was the bash RPM built, and what is its installed size?

rpm -q bash --queryformat '%{BUILTHOST} %{INSTALLSIZE}'

Has the pam package changed since it was installed?

~~rpm~~ rpm -V pam

Which installed packages have "gnome" in their names?

~~rpm -qa | grep gnome~~

Which RPM provides /etc/inittab?

rpm -qf /etc/inittab

Which RPM provides /etc/hosts? Why?

rpm -qf /etc/hosts

What was the last changelog entry for your kernel?

rpm -qf --changelog kernel | head

What are the differences among the following commands?

- a. # **rpm -ivh package file** - installs
- b. # **rpm -Uvh package file** - upgrades - will remove old
- c. # **rpm -Fvh package file** - freshens - will replace only if not already newer

## 2. RPM signatures

Practice checking the signature and integrity of an RPM package file of your choosing from your CD-ROM or from server1.

Import Red Hat's GPG key to RPM's system-wide keyring. The key can be found on first CD or /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release.

Check the signature of some original RPMs from the NFS mount or CD.



Create a corrupted RPM, and then verify it. Start by copying an RPM file to `/tmp`, then use the **cat** command to append some extraneous data to the end of the file.

## Sequence 2: Connecting to a private repository

Scenario: You are asked to connect your system to the private repository located on `server1`.

Deliverable: A system configured to use the repository located on `server1`

Instructions:

1. Create a file `/etc/yum.repos.d/server1.repo` pointing to a repository with the name `GLS` located on the url `http://server1.example.com/pub/gls/RPMS`. Make sure you set the repository to be enabled.

## Sequence 3: Installing new packages using yum

### Instructions:

1. Use yum to list all packages containing 'rhce-ts' in their name.
2. Install the package you just found in the previous step.



## Sequence 4: Updating software using yum

### Instructions:

1. Use **yum** to check if there are updates available for your system.  
  
Replace your existing `/etc/yum.repos.d/server1.repo` file by downloading an updated copy from the URL `ftp://server1.example.com/pub/gls/server1.repo`. This will point yum to additional repositories containing the base Red Hat Enterprise Linux packages and available updates to those packages.
2. Select one package from the previous step and update it.
3. Now install all available updates for your system.

## Sequence 1 Solutions

1. What files are in the `initscripts` package?

```
rpm -ql initscripts
```

On what host was the `bash` RPM built, and what is its installed size?

```
rpm -qi bash
```

Has the `pam` package changed since it was installed?

```
rpm -V pam
```

Which installed packages have "gnome" in their names?

```
rpm -qa | grep gnome
```

Which RPM provides `/etc/inittab`?

```
rpm -qf /etc/inittab
```

`rpm -qf` requires the full path to the file.

Which RPM provides `/etc/hosts`? Why?

```
rpm -qf /etc/fstab
```

No RPM provides `/etc/hosts` because this file is created by Anaconda during installation

What was the last changelog entry for your kernel?

```
rpm -q --changelog kernel | head
```

What are the differences among the following commands?

a. `# rpm -ivh package file`

b. `# rpm -Uvh package file`

c. `# rpm -Fvh package file`

a. `rpm -i` installs the package without replacing earlier versions. If the old and new versions have files in common (like most RPMs) this command will fail.

b. `rpm -U` installs the new package and removes all earlier versions.

c. `rpm -F` only installs the new package if earlier versions exists. The old versions are removed.

2. Import Red Hat's GPG key to RPM's system-wide keyring.

```
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

Check the signature of some original RPMS from the NFS mount or CD.

a. `# mkdir /mnt/server1`

- b. # mount server1:/var/ftp/pub /mnt/server1
- c. # cd /mnt/server1/Server
- d. # rpm -K httpd-version.i386.rpm

Create a corrupted RPM then verify it.

- a. # cp /mnt/server1/Server\  
                                        /httpd-version.i386.rpm /tmp
- b. # cat /bin/date >> /tmp/httpd-version.i386.rpm
- c. # rpm -K /tmp/httpd-version.i386.rpm

This command should fail.



## Sequence 2 Solutions

1. Create the file `/etc/yum.repos.d/server1.repo` with the following content:

```
[GLS]
name=Private classroom repository
baseurl=http://server1.example.com/pub/gls/RPMS
enabled=1
gpgcheck=0
```

Make sure you have configured the repository correctly by issuing the command: **yum list rhce-ts**.

## Sequence 3 Solutions

1. To list all packages containing 'rhce-ts' in their name you could issue the command: **yum list '\*rhce-ts\*'**
2. To install the package rhce-ts you could issue the command **yum install rhce-ts**

When **yum** asks for confirmation enter **y**.

## Sequence 4 Solutions

1. Use **yum** to check if there are updates available for your system.

Replace your existing `/etc/yum.repos.d/server1.repo` file by downloading an updated copy from the URL `ftp://server1.example.com/pub/gls/server1.repo`. This will point yum to additional repositories containing the base Red Hat Enterprise Linux packages and available updates to those packages.

To find out if there are updates available for your system use the command: **yum check-update**

2. To update only a specific package you can use **yum update *package-name***
3. To install all available updates for your system issue the command: **yum update**











# Unit 3

## Kernel Services



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[training@redhat.com](mailto:training@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Understand the purpose and organization of the kernel
- Understand how to load and configure kernel modules.
- Know how to configure the kernel using /proc and sysctl
- Explore hardware devices available on the system



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



# The Linux Kernel

- The kernel constitutes the core part of the Linux operating system.
- Kernel duties:
  - System initialization: detects hardware resources and boots up the system.
  - Process scheduling: determines when processes should run and for how long.
  - Memory Management: allocates memory on behalf of running processes.
  - Security: Constantly verifies filesystem permissions, SELinux contexts and firewall rules.
  - Provides buffers and caches to speed up hardware access.
  - Implements standard network protocols and filesystem formats.
- Documentation available in the `kernel-doc` RPM package



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Linux kernel is the core of the Red Hat Enterprise Linux operating system. It is the kernel's responsibility to control hardware, enforce security and allocate resources such as CPU and memory. Most of the operating system is not Linux but rather a collection of applications that make use of the facilities provided by the linux kernel. In particular, the Free Software Foundation's ([fsf.org](http://fsf.org)) GNU project provides a great deal of utilities and core libraries that have made linux based operating systems possible.

Kernel code is divided into core functionality such as scheduling and memory management and kernel modules for device drivers, filesystems, and other code that may not be appropriate for all systems.

Kernel documentation is provided with the kernel source code and is distributed by Red Hat in the kernel-doc RPM. This documentation can be found under `/usr/share/doc/kernel-doc-*/Documentation`. Unfortunately, the quality of this documentation is somewhat mixed. While some files are kept current, others languish, not seeing major updates since the 2.2 version of the linux kernel.



## Kernel Images and Variants

- Architectures supported: x86, x86\_64, IA64/Itanium, PowerPC64, s390x.
- Three kernel versions available for x86:
  - Regular: one or more processors but 4GB of RAM or less
  - PAE: multiple processors and up to 64G of RAM
  - Xen: needed for virtualization
- Kernels always installed under `/boot/vmlinuz-*`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The general idea behind providing a variety of kernel versions is that the more features are built in, the more overhead it will bring. For 32bit x86, there is a great deal of difference between the kernels.

The standard kernel is usually installed by default. It support multiple processors, a feature known as Symmetric MultiProcessing (SMP). Memory support is limited to 4 GB physical memory though the amount of *virtual* memory could be larger through the use of swap space. The memory limit per process is 3 GB.

The hugemem kernel adds total memory support of up to 64 GB of RAM and uses the 4/4 memory split. This means that processes and the kernel will both be able to use up to 4 GB of physical memory. This comes at a cost, however: all switches from user to kernel space will require the address space to be remapped and this can hurt performance significantly.

The Xen kernel, (i.e. the kernel-xen RPM package) contains the Xen-enabled kernel for both the host and guest operating systems as well as the hypervisor. Xen is a virtual machine that can securely run multiple operating systems in their own sandboxed domains.

cd /proc  
less cpuinfo  
PAE - physical address extension

## Kernel Modules

- Modules are small kernel extensions that may be loaded and unloaded at will
- Can implement drivers, filesystems, firewall, and more
- Are located under `/lib/modules/$(uname -r) /`
- Compiled for a specific kernel version and are provided with the kernel RPM.
- Third party modules may be added



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A kernel module is an optional portion of kernel code that may be loaded after the kernel has been initialized. Only a few modules that are essential to all systems are included directly into the kernel. Dynamic modules that are needed at boot time are loaded by grub into the initial ram disk (initrd). Other modules may be loaded as needed later and are found in the `/lib/modules/` directory.

Modules are used for several reasons:

- Reduced memory footprint: memory is not used for drivers that are not required.
- Flexibility: modules may be added to the system after it has been installed. These modules are often called third-party drivers.
- Maximizing uptime: a module may be unloaded and reloaded as many times as wanted with no reboot.



# Kernel Module Utilities

- **lsmod** provides a list of loaded modules
- **modprobe** can load and unload modules
- **modinfo** displays information about any available module
- `/etc/modprobe.conf` used for module configuration:
  - Parameters to pass to a module whenever it is loaded
  - Aliases to represent a module name
  - Commands to execute when a module is loaded or unloaded



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Modules may be loaded or unloaded using the **modprobe** utility. Loading a module is as easy as:

```
modprobe usb_storage
```

Modprobe, as opposed to other tools such as **insmod** will also load dependencies automatically and it will figure out where the module is located on the filesystem.

To remove a currently loaded module, use **modprobe -r**:

```
modprobe -r usb_storage
```

Note that a module may only be removed if it is not currently in use.

The **lsmod** utility provides a list of all loaded modules, along with the corresponding size and usage count.

```
lsmod | grep usb_storage
usb_storage 72609 1
```

The **modinfo** command can be passed a module name or filename. It will display the associated information, such as the author's name, license, description, module version, dependencies, parameters, etc.

```
modinfo usb_storage
filename:
/lib/modules/2.6.18-1.2747.el5/kernel/drivers/usb/storage/usb-storage.ko
license: GPL
description: USB Mass Storage driver for Linux
author: Matthew Dharm
srcversion: 2AA118E385318B2C39E10D3
depends: scsi_mod
vermagic: 2.6.18-1.2747.el5 SMP mod_unload 686 REGPARM
4KSTACKS gcc-4.1
parm: delay_use:seconds to delay before using a new device (uint)
```

The `/etc/modprobe.conf` configuration file contains persistent settings that apply to modules commonly loaded on the system. Existing settings were added at install time, but new lines may also be added as needed:



```
alias eth0 airo
alias snd-card-0 snd-intel8x0
options snd-intel8x0 index=0
```

## Managing the initrd Image

- The initial RAM disk provides modules loaded early in the boot process.
- This file is located under `/boot/initrd-$(uname -r).img`
- Extra modules sometimes need to be added due to:
  - New hardware added to the system. i.e. SCSI controller
  - New features needed such as USB devices.
  - Module needs to load automatically at boot time.
- Use **mkinitrd** and the **--with** option to rebuild with an extra module:

```
mkinitrd --with=module_name /boot/initrd-$(uname -r).img \
$(uname -r)
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The initial RAM disk (also called initrd) provides the system with a series of modules that are not provided by the kernel image but which are still critical to boot the system correctly. These modules are usually associated with storage devices and filesystems, but may support other features and hardware peripherals.

The initrd file is located under `/boot/initrd-$(uname -r).img`. Although the filename typically uses the format above, the name may be modified as long as the Grub configuration file, `/boot/grub/grub.conf` refers to the right filename.

The initial RAM disk sometimes needs to be rebuilt due to new hardware (such as a new SCSI or SATA controller), new features required early during the boot process, or in order to load a module automatically. To rebuild the initrd file, the **mkinitrd** command is provided. Several options are supported, such as **--with**, which makes it possible to list modules that should be forced into the initial RAM disk. A typical command would look like:

```
mkinitrd --with=usb_storage /boot/initrd-$(uname -r).img $(uname -r)
```

where **\$(uname -r)** is replaced with the current kernel version.



# Accessing Drivers Through /dev

*dev/res*

- Files under /dev used to access drivers
- Reading from and writing to those files are valid operations:
  - Read from serial port: `cat /dev/ttyS0`
  - Write to serial port: `echo "Message" > /dev/ttyS0`
- Three file attributes determine which driver to access:
  - Device type (character or block)
  - Major number
  - Minor number



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Device nodes provide access points for device drivers and map standard file permissions onto that access. The two types of device nodes are block and character. A quick description of these would be that block devices handle data storage while character devices are more suitable for data streams. A more precise description would be that block devices use kernel buffered IO and character devices do not use buffers.

Reading from a serial port, for instance, can be easily accomplished using the following command:

```
cat /dev/ttyS0
```

Writing a word to the same serial port, on the other hand, would require:

```
echo "Message" > /dev/ttyS0
```

Device special files located under /dev are all tagged with three attributes: the device type (character or block), the Major number and the Minor number. The `ls -l` command can be used to display those attributes:

```
ls -l /dev/ttyS0
crw-rw-r-- 1 root uucp 4, 64 Feb 4 16:24 /dev/ttyS0
```

In the previous example, the 'c' at the beginning of the line indicates a character device. Notice that there is no file size: this field was replaced with 4, 64, which indicates a major number equal to 4 and a minor number equal to 64. The major number determines which driver to access, whereas the minor number allows the driver to differentiate between similar physical devices.

Similarly, a block device such as a USB disk would look as follows:

```
ls -l /dev/sda1
brw-r----- 1 root disk 8, 1 Feb 4 16:12 /dev/sda1
```

*major & minor #s  
what kernel sees*



# Device Node Examples

- Block Devices
  - `/dev/hda`, `/dev/hdc` - IDE hard disk, CDROM
  - `/dev/sda`, `/dev/sdb` - SCSI, SATA, or USB Storage
  - `/dev/md0`, `/dev/md1` - Software RAID
- Character Devices
  - `/dev/tty[0-6]` - virtual consoles
  - `/dev/null`, `/dev/zero` - software Devices
  - `/dev/random`, `/dev/urandom` - random Numbers



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

IDE drives are labeled `/dev/hd*`. `/dev/hda` is the master drive on the first IDE controller. `/dev/hdb` is the slave on the first controller. `/dev/hdc` is the master on the second controller and `/dev/hdd` is the slave on the second controller. Most IDE based systems use `/dev/hda` as the primary hard disk with the CD/DVD drive connected as `/dev/hdc`. If an additional drive is added, it will normally be `/dev/hdd` to avoid overloading the first IDE controller.

SCSI storage devices can be a number of different things: plain SCSI hard disks, Serial ATA, SAN storage devices, USB devices. The key to understanding this is that SCSI is a protocol, not a type of device. So even with something like a USB storage device, developers chose to stick with an established standard for the interface. One particular challenge with SCSI devices is how they are named: which drive will be `/dev/sda`, which is `/dev/sdb` and such. The naming of devices occurs in the order the devices are discovered. There is no guarantee that the drive that is `/dev/sdb` now won't become `/dev/sdc` after a reboot. To get around this issue, various labeling schemes are used for filesystems and swap space to avoid addressing the device directly through the device node.

One common class of character device is the terminal. Basically, whenever you're working at a shell, you're using a terminal device. The command **who am i** can be used to check which device you're using. Using just **who** will show what terminals are in use.

The virtual consoles provide text logins through the kernel's VGA driver and are identified as `/dev/tty` followed by a number. There are normally six virtual consoles that provide logins.

Pseudo terminals are used by programs like **gnome-terminal** and the ssh server. These terminals are dynamically created by a special virtual filesystem, `devpts`, under the `/dev/pts/` directory.

## Managing /dev With udev

- udev manages files stored under /dev/
- Files only created if corresponding device is plugged in
- Files are automatically removed when device is disconnected
- udev statements under /etc/udev/rules.d/ determine:
  - Filenames
  - Permissions
  - Owners and groups
  - Commands to execute when a new device shows up



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Linux distributions used to ship with thousands of files under the /dev directory. Although this worked, it certainly was not elegant to provide every single device special file known to mankind even if the corresponding device would never be added to your system.

A newer, better, and more efficient system was then added to replace static files under /dev. This system is called *udev*. It consists in a series of utilities and configuration files which provide rules that apply whenever a device is connected to the system and detected by the kernel.

udev makes it possible to create or remove files on the fly, when the corresponding device is plugged or disconnected. It also lets system administrator add rules in order to modify default names and permissions used under /dev.

Rules are located under /etc/udev/rules.d/. Red Hat Enterprise Linux ships with a default set of rules that should work in most cases. We suggest adding a new file in order to include new rules under udev.

USB is detected as SCSI device



## Adding Files Under /dev

- The right way to add a /dev entry involves udev:

- Create a new file under /etc/udev/rules.d/
- Insert a statement such as:

```
KERNEL=="sda", NAME="usbkey" , SYMLINK="usbstorage"
```

- This creates a device file named `usbkey` and a symlink named `usbstorage` next time `/dev/sda` gets plugged.
- Files can be added manually with **mknod**:

```
mknod /dev/usbdevice b 8 0
```

- **mknod** not persistent!



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Adding a device special file under /dev can be done in two ways: using **mknod** or by adding a new rule to udev.

The udev method involves adding a file under /dev/udev/rules.d/. This file should contain one or several statements that comply with the udev syntax. A statement such as the following would replace the default /dev/sda file with /dev/usbkey. A symbolic link called /dev/usbstorage would also be created. All these files would get automatically created when the device gets plugged in, and removed when disconnected.

Alternatively, a file can be quickly created under /dev using the **mknod** command. This would require knowledge about the device type, major and minor numbers and work as follows:

```
mknod /dev/usbdevice b 8 0
```

The MAKEDEV utility can also be used as an easier replacement to **mknod**. In both cases, the new file would not persist after a reboot.



## Kernel Configuration With /proc

- /proc used to get or set kernel configuration
- Virtual filesystem: files not stored on hard disk
- Entries not persistent: modifications get reinitialized after a reboot
- Used to display process information, memory resources, hardware devices, kernel memory, etc.
- Can be used to modify network and memory subsystems or modify kernel features
- Modifications apply immediately



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

/proc is a virtual filesystem that provides detailed information about the kernel, hardware and running processes. Most files under /proc can be viewed with the **cat** command, though a couple use null characters to separate contents and are better viewed with the **strings** command. There isn't any real consistency to the format of data available through /proc. Some entries are easy to read and interpret, some are only useful to kernel developers and others are clearly not meant to be read by a human.

*Most files in /proc*

# /proc Examples

- Read-only files:
  - /proc/cpuinfo
  - /proc/1/\*
  - /proc/partitions
  - /proc/meminfo
- Read-Write entries under /proc/sys/:
  - /proc/sys/kernel/hostname
  - /proc/sys/net/ipv4/ip\_forward
  - /proc/sys/vm/drop\_caches
  - /proc/sys/vm/swappiness



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Some Interesting /proc entries:

|                                   |                                                                                                      |
|-----------------------------------|------------------------------------------------------------------------------------------------------|
| /proc/<PID>                       | Information on running processes ( <b>ps</b> , <b>top</b> )                                          |
| /proc/cmdline                     | Boot time options                                                                                    |
| /proc/cpuinfo                     | processor information                                                                                |
| /proc/mdstat                      | software RAID information ( <b>mdadm</b> )                                                           |
| /proc/meminfo                     | system memory usage ( <b>free</b> , <b>vmstat</b> )                                                  |
| /proc/swaps                       | system memory usage ( <b>free</b> , <b>vmstat</b> )                                                  |
| /proc/modules                     | dynamically loaded modules ( <b>lsmod</b> )                                                          |
| /proc/mounts                      | mounted filesystems ( <b>mount</b> )                                                                 |
| /proc/net                         | network activity and configuration ( <b>ifconfig</b> , <b>netstat</b> )                              |
| /proc/partitions                  | block devices known to the kernel                                                                    |
| /proc/version                     | version of the linux kernel ( <b>uname</b> )                                                         |
| /proc/sys/kernel/<br>hostname     | System hostname                                                                                      |
| /proc/sys/net/<br>ipv4/ip_forward | IP Forwarding (on or off)                                                                            |
| /proc/sys/vm/<br>drop_caches      | Writing a 1 forces the kernel to free up some memory from caches.                                    |
| /proc/sys/vm/<br>swappiness       | Indicates how aggressively memory will be swapped out to the swap device (number between 0 and 100). |



# sysctl : Persistent Kernel Configuration

- **sysctl** adds persistence to `/proc/sys` settings
- Statements added to `/etc/sysctl.conf` automatically reflected under `/proc` after a reboot.
- Configuration maintained or monitored using the **sysctl** command:
  - List all current settings: **sysctl -a**
  - Reload settings from `sysctl.conf`: **sysctl -p**
  - Set a `/proc` value dynamically: **sysctl -w net.ipv4.ip\_forward=1**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Kernel parameters provide a mechanism to adjust the functioning of the linux kernel. Generally speaking, whenever a kernel developer selects an arbitrary constant or implements functionality that may not be generally desired, a `sysctl` will be made available to adjust it. Because of this, a great many `sysctls` exist, many of which are not documented in a way useful to an administrator. This is generally okay, because there's a good chance that the parameter would not be useful anyway. Commonly useful parameters will be documented online, in `kernel-doc` or in this and other Red Hat courses.

The `sysctl` command can be used to view and set kernel parameters. To get a list of all parameters and their values:

```
sysctl -a
```

To set a parameter, `net.ipv4.tcp_syncookies` in this case, we could use:

```
sysctl -w net.ipv4.tcp_syncookies=1
```

To make this setting permanent, we would want to add the parameter to `/etc/sysctl.conf`.

Once this is done, the new configuration file could be synchronized with the kernel by using:

```
sysctl -p
```

man 5 sysctl.conf



## Exploring Hardware Devices

- A snapshot of all connected devices is maintained by HAL: Hardware Abstraction Layer
- **hal-device** lists all devices in text mode.
- **hal-device-manager** displays all devices on a graphical window.
- **lspci** and **lsusb** list devices connected to the PCI and USB buses, respectively.
- The `/proc` and `/sys` filesystems also contain bus and device specific information.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Hardware devices can be monitored in a number of different ways. The `/proc` filesystem has historically been the main method, with files such as `/proc/devices`, `/proc/interrupts`, `/proc/iomem` and `/proc/ioports`.. Buses such as the PCI and USB bus are also exposed through the `/proc/bus/` directory.

To make the `/proc` entries more readable, utilities such as **lspci** and **lsusb** are also provided.

More recently, however, a new layer has been provided to expose hardware information: HAL (Hardware Abstraction Layer). HAL continuously maintains a snapshot of all hardware devices currently connected to the system. This snapshot may be monitored in text mode using the **hal-device** command, or in graphical mode with the **hal-device-manager** application.

Much of the information provided through HAL can also be accessed from the `/sys` filesystem.

# Monitoring Processes and Resources

- Information available under `/proc/` can be hard to understand.
- Interfaces are available to format the data and make it more accessible:
  - Memory: **free**, **vmstat**, **swapon -s**, **pmap**
  - Processes: **ps**, **top**, **gnome-system-monitor**
  - Kernel state: **uname**, **uptime**, **lload**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

There are a huge number of interfaces that collect and report information from the kernel and cover almost all the functionality of the kernel. Often you'll want to focus on using these tools instead of the raw data found under `/proc` and `/sys` filesystems.

The linux virtual memory system is quite complicated and has a number of tricks up it's sleeve to optimize memory utilization and particularly memory sharing. This makes getting a complete idea of how memory is actually being consumed very difficult to get a grasp on.

When viewing the output of `ps`, two numbers on memory utilization stand out, RSS (the amount of resident memory) and VSIZE (the virtual memory size). VSIZE is not an accurate account of memory usage since the difference between VSIZE and RSS reflects memory space promised to a program, but not yet referenced. Normally this indicates libraries that have been included but are mostly unused. The RSS represents memory actually mapped to the program, but even this isn't a good account of memory consumption since a majority of this memory is most likely shared with other instances of this or other processes. If you peek into `/proc/<PID>/maps` you can see exactly how the memory space has been set up, but even the figures that you could calculate from there are likely misleading because even writable private data could still be shared via the copy-on-write mechanism.

A complete grasp on memory consumption is probably out of our grasp, but the most useful numbers are easy to get at with the **free** and **vmstat** commands. **free** can be used to track your current memory consumption and **vmstat** can be used to track it over time. Particularly with swap utilization, it's not how much you use, but how *often* you use it that counts.











## Lab 3

# Configuring the kernel

---

Goal:                      Develop skills tuning the `/proc` filesystem.

                             Gain some experience working with device special files and modules.

                             Use the tools available to explore hardware resources.

Estimated Duration:    45 minutes

System Setup:

Lab Setup:

Situation:



## Sequence 1: Turning off ping responses

**Scenario:** You want to reduce the exposure of a critical system. One of your strategies is to “hide” it from easy discovery by ICMP ECHO requests.

**Deliverable:** A system that does not respond to **ping**.

**Instructions:**

1. Configure your system, so that it does not respond to any **ping** request. This configuration should survive a reboot.

Hint: Install the `kernel-doc` package and check the kernel documentation on `/usr/share/doc/kernel/Documentation/networking/ip-sysctl.txt`.

2. MANDATORY CLEANUP

- a. Comment out or remove `net.ipv4.icmp_echo_ignore_all=1` from `/etc/sysctl.conf`
- b. Reboot the system.

This is to prevent other things from breaking during the week and help preserve your and your instructor's sanity.

## Optional Sequence 2: Creating a file persistently under /dev/

**Scenario:** You want to make sure the /dev/myusbdisk filename is available after a reboot and can be used to mount a USB device.

**Deliverable:** A system that provides /dev/myusbdisk automatically after a reboot.

**System Setup:** System running in runlevel 5.

### Instructions:

1. Modify the udev subsystem in such a way that /dev/myusbdisk gets automatically created at boot time.
2. Reboot the system.  
  
Plug a USB key to your system and verify that you now have a file named /dev/myusbdisk.
3. **MANDATORY CLEANUP:** Remove the file you have created under /etc/udev/rules.d/ and unplug the USB device.

## Sequence 3: Exploring processes, hardware and memory resources

Scenario: You want to determine what processes are running on your system, which hardware devices are available, and how much RAM is left.

Deliverable:

System Setup: System running in runlevel 5.

Instructions:

1. Determine the top-three processes with the largest memory footprint.
2. Determine the top-three processes with the largest cpu usage.
3. Start a window in which, using **vmstat**, a memory snapshot will be obtained every 5 seconds.  
  
In parallel, start a memory-intensive application and observe the results.
4. Determine what network card (brand and/or model) is currently connected to your system.

top, for mem

pfw (w)

Intel 82540EM Gigabit  
Ethernet Controller

PCI



## Sequence 1 Solutions

1. Configure your system, so that it does not respond to any **ping** request.
  - a. Check the present value of `/proc/sys/net/ipv4/icmp_echo_ignore_all`  
  

```
cat /proc/sys/net/ipv4/icmp_echo_ignore_all
```

It should be currently set to zero which means your system will respond normally to pings.
  - b. Change the value of `/proc/sys/net/ipv4/icmp_echo_ignore_all` to a 1 which will prevent other hosts from successfully pinging your host while not affecting your ability to ping them. Verify your work.  
  

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
cat /proc/sys/net/ipv4/icmp_echo_ignore_all
```
  - c. Now test pinging `server1.example.com`. Pressing *Ctrl-C* will stop the **ping** command and display some statistics for you. You should have been able to ping `server1`.
  - d. Next have someone else try pinging your station. They should not receive any responses back from your system. Alternatively, try to ping your own network address. This should not work either.
  - e. Now reboot your system and try to ping your station again. What happened? Why?
  - f. Remember that changes to the `/proc` filesystem are temporary and if you want them to persist across reboots you need to put an entry in `/etc/sysctl.conf`. Edit `/etc/sysctl.conf` and put the following line at the bottom:  
  

```
net.ipv4.icmp_echo_ignore_all=1
```
  - g. To activate this change run:  
  

```
sysctl -p
```
  - h. Check the value in `/proc`. If it is not set to a 1 then recheck the previous two steps. Next reboot your system and check the value in `/proc` again.

## 2. MANDATORY CLEANUP

- a. Comment out or remove `net.ipv4.icmp_echo_ignore_all=1` from `/etc/sysctl.conf`
- b. Reboot the system.

This is to prevent other things from breaking during the week and help preserve your and your instructor's sanity.

## Optional Sequence 2 Solutions

1. Modify the udev subsystem in such a way that `/dev/myusbdisk` gets automatically created at boot time.

Create a file named `/etc/udev/rules.d/99-usb.rules` and insert the following statement in it:

```
KERNEL=="sdb1", NAME="myusbdisk"
```

*Note:* Systems with IDE harddrives use `sda` for the first USB disk.

2. Reboot the system:

```
init 6
```

Plug a USB key to your system and verify that you now have a file named `/dev/myusbdisk`.

```
ls -l /dev/myusbdisk
```

3. **MANDATORY CLEANUP:** Remove the file you have created under `/etc/udev/rules.d/` and unplug the USB device:

```
rm /etc/udev/rules.d/99-usb.rules
```



## Sequence 3 Solutions

1. Determine the top-three processes with the largest memory footprint.

```
top
```

Type "P" to sort processes in order of decreasing memory usage. The three processes at the top of the list are the three largest.

2. Determine the top-three processes with the largest cpu usage.

```
top
```

Type "M" to sort processes in order of decreasing CPU usage. The three processes at the top of the list are the three busiest processes.

3. Start a window in which, using **vmstat**, a memory snapshot will be obtained every 5 seconds.

```
vmstat 5
```

In parallel, start a memory-intensive application and observe the results.

```
cat /dev/hda > /dev/null
```

4. Determine what network card (brand and/or model) is currently connected to your system. You might first want to take a look at the **lspci** command.

The **hal-device-manager** will also display more information about the network interface card.











# Unit 4

## System Services



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[training@redhat.com](mailto:training@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Understand the importance of time synchronization
- Configure System Logging
- Setup the X Window System
- remotely administer the system
- Automate tasks with cron
- Configure printing



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Network Time Protocol

- Workstation hardware clocks tend to drift over time without correction
- Many application require accurate timing
- Time synchronization makes system logs easier to analyze
- NTP counters the drift by manipulating the length of a second
- NTP clients should use three time servers
- Config file: `/etc/ntp.conf`
- Config tool: **system-config-date**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The PC hardware clock is not accurate enough for many applications. It tends to drift over time. Many applications require exact timing and may react poorly if the clock is reset suddenly to a new time.

NTP counters the drift by modifying the length of a second, much like tuning the pendulum of a old fashioned clock. If the system's time is behind the average of the time servers the second is made shorter so that the system clock races towards the correct time. Thus the time difference is reduced gently without disturbing other applications. However if the time differs to greatly, NTP ceases to work. In this case the clock must be reset manually with **ntpdate**.

Having three central time servers allows clients to reject bogus synchronization messages if one of the servers' NTP daemons or clocks malfunctions. It is possible for a client to synchronize with fewer time servers if necessary but it is less secure.

NTP is configured in `/etc/ntp.conf`. For simple client configurations however it is often enough to use the graphical configuration tool **system-config-date** which is automatically run during firstboot .

```
server 192.168.0.1
server 192.168.0.2
server clock.example.com
driftfile /var/lib/ntp/drift
```



# System Logging

- Centralized logging daemons: **syslogd**, **klogd**
- Log file examples:
  - `/var/log/dmesg`: Kernel boot messages
  - `/var/log/messages`:#Standard system error messages
  - `/var/log/maillog`: Mail system messages
  - `/var/log/secure`: Security, authentication, and xinetd messages
- Application log files and directories also reside in `/var/log`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A variety of log files are maintained by the system, an understanding of which is often vital for troubleshooting system problems:

`/var/log/dmesg` -- This logfile is written upon system boot. It contains messages from the kernel that were raised during the boot process.

`/var/log/messages` -- This is the standard system logfile, which contains messages from all your system software, non-kernel boot issues, and messages that go to dmesg. Readable only by root.

`/var/log/maillog` -- This logfile contains messages and errors from your sendmail. Readable only by root.

`/var/log/secure` -- This logfile contains messages and errors from security-related systems such as login, tcp\_wrappers, and xinetd. Readable only by root. Very useful in detecting and investigating network abuse.

There are also various other system logfiles that store information from other applications (i.e. Apache, Squid, etc.) that also may be found under `/var/log`.

# syslog Configuration

- **syslog** System V initialization script in `/etc/rc.d/init.d` controls both the **syslogd** and the **klogd** daemons
- `/etc/syslog.conf`
  - Configures system logging
- `/etc/sysconfig/syslog`
  - Sets switches used when starting **syslogd** and **klogd** from the System V initialization script



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Enterprise Linux provides a central system logging facility that allows all applications to store debugging information, errors and messages in a central, manageable place. System logging is provided by **syslogd**. **klogd** intercepts kernel messages and provides them to **syslogd**. **syslogd** is configured in `/etc/syslog.conf`.

Messages can be logged to files, broadcast to connected users, written to the console, or even transmitted to remote logging daemons across the network. Messages for system logging all have an associated severity. You may choose not to log less-severe messages, or to log severe messages in a separate file. On most systems however, the default settings are adequate.

By default, messages of emergency or higher (more severe) are broadcast to all users, and most other messages are written to `/var/log/messages`, which is where you should look for non-kernel boot errors, error messages from most application-level services, such as automount, login services, etc. After system boot, kernel messages are also written to this file.

## *Log Format:*

Each entry in the system log file has four main entries:

The date and time of the message.

The hostname from whence the message came. This field is important when logging across a network to a centralized log host.

The name of the application or subsystem from whence the message came; for example, kernel, vsftpd, etc. This may include the process identifier.

The remainder of the line (following the colon) is the actual message itself.

Repeated log messages will be marked as such.



# XOrg: The X11 Server

- Foundation for the Red Hat Enterprise Linux graphical user interface(GUI)
- Open Source implementation of X11
- Client / Server Architecture
- Core server with dynamically loaded modules
  - drivers: ati, nv, mouse, keyboard, etc.
  - extensions: dri, glx, and extmod
- Font Rendering
  - Native server: xfs
  - Fontconfig/Xft libraries



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The X Window System (also referred to as “X” or “X11”) is the foundation for the graphical user interface(GUI) on Red Hat Enterprise Linux. The X Window System is maintained by the X Consortium at <http://www.X.org> and creates the reference implementation of X under an open source license. The XOrg project (<http://xorg.freedesktop.org>) adds hardware drivers for a variety of video cards and input devices, along with several software extensions to manage the visual representation of data.

It is important to understand X's relationship to what you see on the screen. X does not define how anything should look or behave. Instead, X focuses on providing a standard way in which applications, called X clients, may display, or “write” on the screen. The X server is the program that speaks through your video hardware. Any application that wants to communicate through the display is an X client, including the **OK** button changing color when clicked with the left mouse button.

XOrg consists of one core server and several dynamically loaded modules. The server core is hardware independent and is extended through the configuration and loading of hardware and X11 extension specific modules and the kernel's module directory. Like the OS kernel, through the combination of unique, hardware specific modules, the X server design remains flexible. In fact, when the server is invoked, it scans and rectifies its configuration against the hardware, seeking the optimum display and input capabilities. To investigate this process, look through the `/var/log/Xorg.0.log` file.

Fonts, or the character sets used to render numbers, letters and other *glyphs* (like dollar or euro symbols) are managed either through the native X server rendering engine, or the libraries of fontconfig/Xft. The native server is actually a separate service, named xfs and must be available to invoke the X server. The Xft rendering engine is implemented within the XOrg core server through dynamically loaded libraries. It is planned to succeed xfs for its flexibility, enhanced rendering capabilities, programming API, and utilities to adjust font rendering in the KDE and GNOME object frameworks. We will discuss a few of these utilities later.

The XOrg X server has native support for several types of fonts, including TrueType and Postscript Type 1. To render all fonts accurately, be sure you have the freetype module defined in the “Module” section of the server configuration(discussed later).

To provide the local X server fonts not shipped with the RHEL distribution, copy them to a directory under



`/usr/share/fonts`, or to the end-user directory `$HOME/.fonts`. The next time an X session starts, the Xft system spawns `fc-cache`, automatically configuring fonts under these directories for most client applications (one notable exception is the OpenOffice Suite which has its own font management system).

By default, `xfs` listens on a Unix domain socket and does not accept remote network connections. To change this behavior, comment out the bottom line of the `xfs` configuration file, `"no-listen = tcp"`. As with any network service, take appropriate security measures at your firewall, or on the font server machine itself with `iptables` or `system-config-security` to limit which clients can connect to it. Network font servers listen on TCP port 7100.

# XOrg Server Configuration

- Typically configured after installation
- Post-install configuration:
  - Best results while in runlevel 3!
  - **system-config-display**
    - options:
      - **--noui**
      - **--reconfig**
  - stored in `/etc/X11/xorg.conf`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

If the X environment was not configured during installation, the utility `system-config-display` is used with the recommended **--noui** and **--reconfig** options. With these options, probed values for the monitor and video card will be used, with a deference toward the greater capabilities of system hardware. For example, if the probed hardware can support a color depth of 24 bits/pixel(TrueColor), then the server will start with this setting. If the probed attempt fails, then typical and conservative values will be set. The resulting video card and display, keyboard, and mouse configuration is stored in

`/etc/X11/xorg.conf` and specifies the hardware components' resources in several sections.

Among them are:

“Server Layout”

defining individual combinations of “InputDevice” and “Screen”. Only one per session is used.

“Module”

defining which hardware and extension specific modules are called when X is started.

“InputDevice”

defining keyboard, mouse, touchscreen, or other form of supported input device.

“Monitor”

defining characteristics and capabilities of the physically attached output hardware.

“Device”

defining which hardware specific driver is used to communicate with the local video card.

“Screen”

defining individual combinations of “Monitor” and “Device” with display properties.

Please Note: X server configuration should be performed in runlevel 3 for best results. Also note that to run an X client to be displayed on a remote system, no local server configuration is necessary.

## XOrg in runlevel 3

- Two methods to establish the environment
  - `/usr/X11R6/bin/xinit`
  - `/usr/X11R6/bin/startx`
- Environment configuration
  - `/etc/X11/xinit/xinitrc` and `~/.xinitrc`
  - `/etc/X11/xinit/Xclients` and `~/.Xclients`
  - `/etc/sysconfig/desktop`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

To start an X session in run level 3, use `xinit` or `startx` from a virtual console shell prompt.

**init** (or **startx**) will pass control of the X session to `/etc/X11/xinit/xinitrc`, unless `~/.xinitrc`, exists.

`xinitrc` seeks to read additional system and user configuration files, including:

Resource files:

`/etc/X11/Xresources` and `$HOME/.Xresources`

Input device configuration files:

`/etc/X11/Xkbmap` and `$HOME/.Xkbmap`,

`/etc/X11/Xmodmap` and `$HOME/.Xmodmap`

**xinitrc** then runs all shell scripts in `/etc/X11/xinit/xinitrc.d`.

`xinitrc` then turns over control of the X session to `~/.Xclients`, if it exists. If not, it turns over control to `/etc/X11/xinit/Xclients`.

**Xclients** reads `/etc/sysconfig/desktop` to determine whether Gnome or KDE is the preferred desktop environment. If unset, or the defined is not installed, `Xclients` will attempt to run a number of other window managers in the following order:

1. Gnome
2. KDE
3. **twm** (plus **xclock**, **xterm**, and **mozilla**, if installed)

In the unlikely event that **Xclients** does not exist, **xinitrc** will go into failsafe mode, starting an **xclock**, **xterm**, perhaps **mozilla**, and finally the **twm** window manager.



# XOrg in runlevel 5

- Environment established by **/sbin/init**
- Environment configuration
  - **/etc/inittab**
    - **/etc/X11/prefdm**
  - **/etc/sysconfig/desktop**
    - DESKTOP defines the window manager
    - DISPLAYMANAGER defines the display manager
  - **/etc/X11/xdm/Xsession**
    - **/etc/X11/xinit/xinitrc.d/\***
    - **~/.xsession** or **~/.Xclients**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

If the default runlevel is set to 5 in **/etc/inittab**, **/sbin/init** will run **/etc/X11/prefdm**. This script invokes the X server and a display manager, set in the file **/etc/sysconfig/desktop**. If no system default is set, it tries **gdm**, then **kdm**, then **xdm**.

By default, the three display managers all use the same startup scripts as part of their operation. When a display manager is first started, the **/etc/X11/xdm/Xsetup\_0** file is run (as root) before the display manager presents a login widget. Then, once the user authenticates, **/etc/X11/xdm/Xsession** is run. This does many of the same things **/etc/X11/xinit/xinitrc** does in **startx**, including running the executables in **/etc/X11/xinit/xinitrc.d**.

**/etc/X11/xdm/Xsession** determines which desktop environment to run. If the user specified one at login through the display manager, it is run. Otherwise, the script checks for either **~/.xsession** or **~/.Xclients**. If all else fails, **/etc/X11/xinit/Xclients** is run as it is for **startx**.

When the user logs out at the end of their session, the X server is restarted by the display manager with a new login window.

# Remote X Sessions

- X protocol communication is unencrypted
- Host-based sessions implemented through the **xhost** command
- User-based sessions implemented through the Xauthority mechanism
- **sshd** may automatically install **xauth** keys on remote machine
  - Tunnels X protocol over secure encrypted **ssh** connection



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The X11 environment was originally designed to run on one host and serve remote client connections. While this capability is still available, because data is transmitted unencrypted it is disabled by default. All three display managers may be configured to host remote X sessions and act as a system authentication agent. For example, a user on one host client may request a display managed session from another host server which in turn provides access to server's system resources. The syntax for making the request is `X -query server`, if an X session is not already running on client, and `X -query server :1` if one instance of X is already running.

To locally display an X client application from a remote host, you must configure your local X server and remote hosts. This technique is useful when you wish to run a GUI client to manage resources or services on a remote system with no console attached (a headless server, as found in most data-centers). A few methods are available, including **xhost**, **xauth**, or forwarded through an SSH connection. In all cases, display access is permitted by the local console and X server session owner.

As its name suggests, **xhost** permits connections on a host to host basis. When access to your display is permitted by this method, *all* redirected X client connections are accepted from the permitted host, regardless of the user who spawns them! If you require host-based X access, issue the command **xhost + trustedhost**, which instructs the local X server that clients from **trustedhost** may have access to the display. Similarly, **xhost - friendlyhost** removes this access permission. The command **xhost +** without a hostname argument disables access restrictions, and is not recommended.

For greater accountability, a user-based mechanism may be preferable. The `$HOME/.Xauthority` file, used in conjunction with the **xauth** command, provides the capability of authenticating remote users who are permitted to use the local display. While certainly more secure than **xhost**, data is similarly passed unencrypted between systems. An even more secure method utilizes *Secure Shell* (SSH).

SSH provides an excellent user-based access mechanism that is encrypted, easy to use and compatible with the **xauth** facility. Using the **xauth** program for display access, SSH will tunnel X authentication information between systems. To achieve this, use **ssh -Y remote-host**.



# SSH: Secure Shell

- encrypted remote shell
- frequently used for remote system administration
- can copy files securely
- can execute commands remotely

```
ssh root@host 'ifconfig eth0'
```

- can tunnel X11 and other TCP based network traffic
- supports key based authentication



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Secure Shell (SSH) is one of the most versatile system administration tools. It allows login and execution of commands on remote systems. It uses strong encryption and host keys as a protection against network sniffing. It is the only network service which is enabled by default and is remotely accessible. The OpenSSH server configuration usually does not require modification.

Apart from logging in remotely you can also use **ssh** to execute a single command remotely. Input and Output channels are tunneled securely to the other side.

```
echo "hello world" | ssh root@host 'cat >/tmp/file'
```

SSH can also be used to tunnel TCP traffic between the SSH server and client:

```
ssh -L 8080:remote-server:80 user@ssh-server
```

This example redirects port 8080 of the local system to port 80 of the remote server. For the remote server, the connection seems to be initiated by the SSH server. So in this example by pointing your webbrowser to <http://localhost:8080> you will access the webpage on `remote-server:80`. The webserver will report the SSH server in it's access log.

The most useful feature of SSH is the ability to authenticate with ssh keys. For that the user has to generate a key pair and put the so called public key into the `~/.ssh/authorized_keys` file on the server. This key is normally protected by a passphrase and managed by the **ssh-agent**. However if the key is used for task automation it must not incorporate a passphrase.

```
ssh-keygen -d
```

Generating public/private dsa key pair.

Enter file in which to save the key (/root/.ssh/id\_dsa): **Enter**

Enter passphrase (empty for no passphrase): **Enter**

Enter same passphrase again: **Enter**

```
cat /root/.ssh/id_dsa.pub | ssh root@server 'cat >>.ssh/authorized_keys'
```



# VNC: Virtual Network Computing

- Allows to access or share a complete desktop over the network
- Uses significantly less bandwidth as pure remote X connections
- Server
  - Individual users can start a VNC server with the command: **vncserver**
  - Runs `$HOME/.vnc/xstartup` upon startup
  - Requires a VNC password which should not be identical to the system password
  - Servers can automatically be started via `/etc/init.d/vncserver`
- Client
  - connects to a remote VNC server with **vncviewer host:screen**
  - Unique screen numbers distinguish between multiple VNC servers running on the same host
  - supports tunneling through SSH: **vncviewer -via user@host localhost:1**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Virtual Network Computing (VNC) is a platform independent way to access a complete desktop over the network. It uses significantly less resources than a traditional remote X11 connection. VNC sessions can be shared by multiple clients and also be used in “view-only” mode for demonstrations. In system administration VNC is most commonly used to administer machines graphically or to monitor installations.

Each user can setup a VNC password and start **vncserver** manually. The desktop is initialized using `$HOME/.vnc/xstartup`.

The system administrator can use `/etc/init.d/vncserver` to automate startup. This System V script reads `/etc/sysconfig/vncservers` and starts all VNC servers specified by `VNCSERVERS`.

Clients can access the server by using **vncviewer host:screen**. `host` is the hostname or IP of the remote server, `screen` the unique screen number assigned to the particular server process.

However this connection is not encrypted. Therefore it is strongly recommended to use the `-via user@host` option. This creates an SSH tunnel. Use `localhost` as the VNC host name in this case.

When a second VNC client connects to the same server the first one is disconnected. The first client can allow multiple connections by specifying the `-Shared` option.

## cron

- Used to schedule recurring events
- Use crontab to edit, install, and view job schedules
- Syntax
  - `crontab [-u user] file`
  - `crontab [-l | -r | -e]`
    - `-l` lists crontab
    - `-r` removes crontab
    - `-e` edits crontab using `$EDITOR`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

*Cron table files* (crontabs) are stored in `/var/spool/cron`, which is not accessible by non-privileged users and generally should not be access by those who do have root privileges In order to access the current crontab, the `crontab` command is used.

The crontab can either be edited in the current directory and installed by using it as an argument to **crontab** or by using the **-e** option.

Events scheduled by cron are run by the `crond` daemon, which must be running on the system.

Fields in a crontab can be separated by any number of tabs or spaces.

Valid field values are as follows:

|              |                                                                   |
|--------------|-------------------------------------------------------------------|
| Minute       | 0-59                                                              |
| Hour         | 0-23                                                              |
| Day of Month | 1-31                                                              |
| Month        | 1-12 or Jan, Feb, Mar, etc.                                       |
| Day of Week  | 0-7, where 0 or 7= Sunday, 1=Monday, etc.; or Sun, Mon, Tue, etc. |

Multiple values may be separated by commas. Also, an "\*" symbol in a field represents all valid values.

The syntax for user crontab files is well documented in section 5 of the man pages.

**man 5 crontab**



# Controlling Access to cron

- Restrict / allow user access to cron
  - `/etc/cron.allow`
  - `/etc/cron.deny`
- Contains usernames to allow / deny access



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## cron access control

- If neither `/etc/cron.allow` nor `/etc/cron.deny` exists only root is allowed to install new crontab files.
- If only `/etc/cron.deny` exists, all users except those listed in `cron.deny` can install crontab files.
- If only `/etc/cron.allow` exists, root and all listed users can install crontab files.
- If both files exist `cron.deny` is ignored.

Note that denying a user through the use of the above files does not disable their installed crontab.



## System crontab Files

- Different format than user crontab files
- Master crontab file `/etc/crontab` runs executables in
  - `/etc/cron.hourly`
  - `/etc/cron.daily`
  - `/etc/cron.weekly`
  - `/etc/cron.monthly`
- `/etc/cron.d/` directory contains additional system crontab files



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The format of `/etc/crontab` and the files in `/etc/cron.d` are different from user crontabs. The sixth field is a username which will be used to execute the command in the seventh field.

A common command in these files is the run-parts shell script. This script takes one argument, a directory name, and invokes all of the programs in that directory (The run-parts script is located in `/usr/bin` and has no online documentation.).

The following is an example `/etc/crontab` file:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/bin:/usr/sbin
MAILTO=root
HOME=/

run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Thus, at 4:02 every morning, all of the executables in the `/etc/cron.daily` directory will be run as root. A default installation's `cron.daily` directory will contain scripts to update the `slocate` and `whatis` databases, to clean up temporary directories, and to perform other housekeeping tasks.

# Daily Cron Jobs

- **tmpwatch**
  - Cleans old files in specific directories
  - Keeps /tmp from filling up
- **logrotate**
  - Keeps log files from getting too large
  - Highly configurable in /etc/logrotate.conf
- **logwatch**
  - provides a summary about system activity
  - reports suspicious messages
  - Configuration file: /etc/log.d/conf/logwatch.conf



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## tmpwatch

**tmpwatch** deletes all files in /tmp that have not been accessed for 240 hours (10 days). It also deletes files in /var/tmp that have not been accessed for 720 hours (30 days).

## logrotate

Left unchecked, system logs will grow until you run out of disk space. **logrotate** rotates log files from different subsystems at predefined intervals, or when they reach predefined sizes, and old logs are optionally compressed.

For example, /var/log/messages is rotated weekly to /var/log/messages.1, with older log files rotated to /var/log/messages.2, etc., optionally compressed.

Configuration is stored in /etc/logrotate.conf, for general settings, and /etc/logrotate.d/subsystem, for subsystem-specific settings. Service specific rotation rules are usually installed by the service's RPM.

## logwatch

Monitoring system logs is an onerous but important task. If you do not properly monitor your logs, you may miss security problems, hardware problems, or software problems. For example, a system may have a runaway maintenance problem every Monday at 5 AM that filled up the filesystem and then promptly cleaned it up again. Only the system log would reveal that there was a problem.

logwatch, installed by default on most Red Hat Enterprise Linux systems, monitors log files, reporting nightly on activity, and, potentially, on any anomalies located. logwatch is highly configurable. It can be programmed to detect most any type of activity. See /usr/share/doc/logwatch-version for information on writing log filters.



# The anacron System

- **anacron** runs **cron** jobs that did not run when the computer is down
  - Assumes computers are not up continually
  - Vital for laptops, desktops, workstations, and other systems that are not up continually
  - Useful for servers that need to be taken down temporarily
- Configuration file: `/etc/anacrontab`
  - Field 1: If the job has not been run in this many days...
  - Field 2: wait this number of minutes after reboot and then run it
  - Field 3: job identifier
  - Field 4: the job to run



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The **anacron** command now runs at boot time. The purpose of the **anacron** command is to run **cron** jobs that would have run had the computer not been down.

This is an essential tool for computers that are not constantly running, such as workstations, desktops, and laptop computers. It can also be useful for servers when they need to go down for an extended period of time. For example, the default configuration of **cron** runs the scripts in `/etc/cron.daily` once a day at 4:02am, according to the `/etc/crontab` file. Among the commands that are run from `cron.daily` is the **mlocate.cron** command, which recreates the **mlocate** database. If the laptop is almost always off at 4:02am, then the **mlocate** database will almost never be updated. For this reason, it is essential that **cron** jobs run on a regular basis. The solution for this is the **anacron** command.

Here is how **anacron** works:

When **cron** runs the **run-parts** command from `/etc/crontab` for `cron.daily`, `cron.weekly`, or `cron.monthly`, the first command to run is **0anacron**. This command sets a timestamp in a file in `/var/spool/anacron` that notes the time that this was last run.

On boot up, the **anacron** command runs. The `anacrontab` file specifies how often the commands in `cron.daily`, `cron.weekly`, and `cron.monthly` should run. If they have not run in this time, then **anacron** waits a few minutes (as specified in the second field of the `anacrontab` file) and then runs the commands, thus ensuring that if the computer was down during the time that **cron** should have run these commands, they are, nonetheless, run.



# CUPS

- uses the Internet Printing Protocol (IPP)
  - allows remote browsing of printer queues
  - based on HTTP/1.1
  - Uses PPD files to describe printers
- Configuration files
  - `/etc/cups/cupsd.conf`
  - `/etc/cups/printers.conf`
- Configuration tools
  - **system-config-printer**
  - Web based on `http://localhost:631`
  - Commandline management with **lpadmin**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Common Unix Printing System (CUPS) supports many advanced features such as printer pooling, automatic client configuration, job redirection and others. The Internet Printing Protocol (IPP) is based on HTTP/1.1. The web based management console of CUPS can be reached at port 631. By default only members of the `sys` group can connect. This console can also be used for configuration.

A mini-LPD service, **cups-lpd**, is available for backward compatibility with older LPRng client systems.

The main configuration file for cupsd is `/etc/cups/cupsd.conf`. This file is very similar in format to the Apache webserver's `/etc/httpd/conf/httpd.conf` configuration file. Documentation in `/usr/share/doc/cups-version/` is also shared out as web pages through the web-based administrative interface. `/etc/cups/printers.conf` is automatically generated by the configuration tools and contains the configuration of the individual printer queues.

**system-config-printer** is a graphical utility that can ease setting up local and remote printers. It has the ability to set the system default printer and can configure CUPS to print to local and remote print queues including IPP, Windows, Novell and Unix LPD queues.

On the command line printer queues can be managed with **lpadmin**. CUPS has both a System V and BSD command line interface. You can use **lp lpr** to print. **lpstat** shows information about the local queues.

*localhost:631*  
*printers web browser*

## End of Unit 4

- Questions and Answers
- Summary
  - System Logging
  - system-config-display
  - Remote Administration tools: **ssh** and **vnc**
  - Task Automation
  - What are the tools to configure cups?



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.







# Lab 4

## System Services

---

Goal: Develop skills using system administration tools and setting up and administering CUPS.

Estimated Duration: 30

## Sequence 1: Using cron

*usr/bin/free /bin/ps*

**Scenario:** You want to know some information about the status of the system every ten minutes today to help investigate some performance issues you have been having. You suspect it might be memory or I/O related and want to keep an eye on those resources.

**Deliverable:** A cron job that runs every ten minutes today between the hours of 0800 (8:00 AM) and 1700 (5:00 PM).

### Instructions:

1. Using **crontab -e** create a cron job that executes **/usr/bin/free** and **/bin/ps** every 10 minutes between 0800 (8:00 am) and 1750 (5:50 pm).
2. How could you send the output from these cron jobs to another e-mail address?  

---
3. Use **mail**, or **mutt** as root to check for email from the cron job you have scheduled.
4. Be sure to delete your cron job when you have received several emails from it.



## Sequence 2: Logging to a centralized loghost

**Scenario:** Your boss thinks it is a great idea to have one central logging host.

Work together with your neighbor. Set up one machine as the log server and the other as a log client

### Instructions:

1. First, on the log server set up syslogd to accept remote messages.
2. On the log client set up syslogd to send messages from the user facility to the log server.
3. Test the new setup by using logger to generate a syslog message. Does the message appear in your neighbor's `/var/log/messages`?

## Sequence 3: Setting up a printer and administering a printer with CUPS.

### Instructions:

1. With **system-config-printer** create a local queue named `lp0` using the generic Postscript printer driver
2. Try to print a file. Since you do not have a printer attached to `/dev/lp0`, this job will be queued forever. Verify the status of the print queue and then delete the print job.

## Sequence 1 Solutions

1. Using **crontab -e** create a cron job that executes **/usr/bin/free** and **/bin/ps** every 10 minutes between 0800 (8:00 am) and 1750 (5:50 pm).
  - a. As root, use the command **crontab -e** to edit your cron file. (If you are not comfortable with vi, export an EDITOR environment variable to set another editor.)
  - b. Enter the following line in your crontab file:  

```
*/10 8-17 * * * /usr/bin/free; /bin/ps
```
2. You can set the MAILTO=*user* variable or use command | **mail -s test user**



## Sequence 2 Solutions

1. First, on the log server set up syslogd to accept remote messages.
  - a. Edit `/etc/sysconfig/syslog`:  

```
SYSLOGD_OPTIONS="-r -m 0"
```
  - b. Restart syslogd:  

```
service syslog restart
```
2. On the log client set up syslogd to send messages from the user facility to the log server.
  - a. Append in `/etc/syslog.conf` the following line:  

```
user.* @192.168.0.X
```
  - b. Restart syslogd.  

```
service syslog restart
```
3. Test the new setup by using logger to generate a syslog message. Does the message appear in your neighbor's `/var/log/messages`?  

```
logger -i -t yourname "This is a test"
```

## Sequence 3 Solutions

1. With **system-config-printer** create a local queue named `lp0` using the generic Postscript printer driver
  - a. As root run **system-config-printer**
  - b. Select **New Printer** and press *Enter*.
  - c. Enter the text: `lp0` in the **Printer Name** field.
  - d. `lp0` in the **Printer Name** field.
  - e. Select Other for the Queue Type
  - f. Select Next and press *Enter*.
  - g. Select `/dev/lp0` and select Next.
  - h. Select Postscript Printer and select Next
  - i. When presented with the screen entitled **Create a New Queue: Name and Type** select Finish and press Enter.
  - j. Select Exit and press Enter. You will be asked to save your changes. Choose Yes and press Enter.
2. Try to print a file. Since you do not have a printer attached to `/dev/lp0`, this job will be queued forever. Verify the status of the print queue and then delete the print job.
  - a. `# lp /root/install.log`
  - b. `# lpstat`  
(Note: you should see one print job active for root with Job number 1)
  - c. Type the command: **cancel 1** to remove the job.
  - d. Type the command: **lpstat** (Note: the job should now be removed)









## Unit 5

# User Administration



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[training@redhat.com](mailto:training@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



# Objectives

Upon completion of this unit, you should be able to:

- Configure user and group accounts
- Modify File ownership and permissions
- Use “Special” permissions SUID / SGID /#Sticky
- Configure Network Users with NIS and LDAP
- Set ACLs



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Adding a New User Account

- Most common method is **useradd**:
  - **useradd** [*options*] *username*
- Running **useradd** is equivalent to:
  - editing `/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/gshadow`
  - creating and populating home directory
  - setting permissions and ownership
- Set account password using **passwd**
- Accounts may be added in a batch with **newusers**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The command-line utility **useradd** provides a simple method for adding new users to the system:

```
[root@stationX]# useradd joshua
```

The above command will add a new account to the machine called **joshua** as well as set up that user's home directory, and create a private group for the user, also called **joshua**. The next step would be to assign **joshua** a password which you can do by simply typing the following command:

```
[root@stationX]# passwd joshua
```

When you need to add several users, you can use the **newusers** command. Create a file, formatted like `/etc/passwd`, that contains the usernames and plain text passwords of the users. The command: **newusers** *yourfile* will create those users. One apparent drawback of using this method is that the users home directories do not get populated with the files from `/etc/skel/`.

```
[root@stationX]# cat myusers
alice:password:500:500:Alice:/home/alice:/bin/bash
bob:Pa55w0rd:501:501:Bob:/home/bob:/bin/bash
[root@stationX]# newusers myusers
```

If you are using MD5 passwords (the default), ensure that **newusers** will generate MD5 passwords, else the users will not be able to login!

`/etc/login.defs` needs an entry like this:

```
MD5_CRYPT_ENAB yes
```



## User Private Groups

- When user accounts are created, a private group is also created with the same name
  - Users are assigned to this private group
  - User's new files affiliated with this group
- Advantage: Prevents new files from belonging to a "public" group
- Disadvantage: May encourage making files "world-accessible"



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Each new user added to a Red Hat Enterprise Linux system also triggers the creation of a new User Private Group, with the same name as the new user. This scheme allows the user's umask to be set to 002, but still restricts write access to files and directories created by the user for other users.

The idea is that if the user has a umask setting of 002, new files and directories will be protected since they will belong to a group that has only one member, namely the user. Of course, the user may belong to other groups and switch to one with the **newgrp** command and/or change the file's group ownership with **chgrp**:

```
[root@stationX]# chgrp groupname filename
```

The drawback is that often users will not do any of these and just make the file world-accessible with **chmod** so that anyone can access it when only a few may need access.

When a system administrator decides to enable multiple users write access to files in a directory, those users are added to a common group. The group ownership on the directory and its files is changed to the common group, and those directories and files are given group-write permission.

When the group set bit is set on the common directory with **chmod g+s**, a file or subdirectory created there by a member of the common group will be owned by that user, but will be assigned group ownership of the common group.

New to Red Hat Enterprise Linux 5 is a **umask** change for normal users. Users no longer get a umask of 002--every user has a umask of 022. Thus if group collaboration is desired, you may want to set the umask in `/etc/bashrc` or `.bashrc` (either in the user's home directory or in `/etc/skel/` so that each new user has the proper umask).



# Modifying / Deleting User Accounts

- To change fields in a user's `/etc/passwd` entry you can:
  - Edit the file by hand
  - Use **usermod** *[options]* **username**
- To remove a user either:
  - Manually remove the user from `/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/gshadow`, `/var/spool/mail`, etc.
  - Use **userdel** **[-r]** **username**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

In addition to hand-editing `/etc/passwd`, you may use **usermod** to change account information. The following options are available:

|                        |                                                                                                                                                                           |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c comment</b>      | Change the comment field. This is often the users full name.                                                                                                              |
| <b>-d home dir</b>     | Change the home directory.                                                                                                                                                |
| <b>-e expire date</b>  | Set date on which the account will expire and be disabled.                                                                                                                |
| <b>-g group</b>        | Change the initial login group                                                                                                                                            |
| <b>-G group, [...]</b> | A comma separated list of supplementary groups for the user. Adding the <b>-a</b> option before the <b>-G</b> option will <i>append</i> the user to the specified groups. |
| <b>-l login name</b>   | Change the login name.                                                                                                                                                    |
| <b>-s shell</b>        | Change the login shell.                                                                                                                                                   |
| <b>-u uid</b>          | Change the login ID.                                                                                                                                                      |
| <b>-p password</b>     | Change the string in the password field. This must be the <i>encrypted</i> password.                                                                                      |
| <b>-L</b>              | Lock the password. This renders the account unusable.                                                                                                                     |
| <b>-U</b>              | Unlock the password.                                                                                                                                                      |

**userdel** deletes a user from the system. It may be prudent to lock the user's account first with **usermod -L**, and to delay deletion of the user's account until you are sure that none of the files in the user's home directory are still needed.

When deleting an account with **userdel**, you should consider using the **-r** option, which recursively deletes the user's home directory. Deleting accounts without deleting the associated home directories may cause issues with file ownership when future users are added to the system.

When adding user access to a group with the **-G** option, you must list all groups to which a user belongs, or include the **-a** option to append the user to the specified groups.

# Group Administration

- Entries added to `/etc/group` and `/etc/gshadow`
  - **groupadd**
  - **groupmod**
  - **groupdel**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

New groups may be created by hand-editing the file `/etc/group` or by using **groupadd**. The basic syntax for **groupadd** is very simple:

```
[root@stationX]# groupadd groupname
```

**groupdel** is used in a similar fashion to remove groups:

```
[root@stationX]# groupdel groupname
```

**groupmod** can be used, among other things, to change the name of a group:

```
[root@stationX]# groupmod -n newname oldname
```

For example, if several users are members of the `employee` group and a root user issues the following command, the group will be changed to `staff` and all the same members will remain:

```
[root@stationX]# groupmod -n staff employee
```

The syntax of `/etc/group` is as follows, one group per line:

```
gurus:x:501:joshua,dax,bryan,chris,heather,jon
```

The first field is the name of the group. The second field is the group password, or an "x" when using shadow passwords. The third field is the unique group ID. The fourth field is a comma-separated list of group members.

In order to avoid using a GID within the range typically assigned to users and their private groups, use the **-r** option:

```
[root@stationX]# groupadd -r groupname
```

This starts the group IDs at 101 and will increase them up to GID 499.

**gpasswd** can be used to define group members in `/etc/group`, group administrators, and to create or change group passwords in `/etc/gshadow`, if desired.



# Password Aging Policies

- By default, passwords do not expire
- Forcing passwords to expire is part of a strong security policy
- Modify default expiration settings in `/etc/login.defs`
- To modify password aging for existing users, use the **chage** command
  - **chage** *[options] username*



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

By default passwords do not expire. This means that it is possible for a user to have the same password indefinitely. This situation is not very secure, because if a password has leaked, or been compromised, it will remain so forever. This can be adjusted in the `/etc/login.defs` file.

The **chage** command is used to set up password aging. You may set the maximum amount of time that a password is considered valid before the system will force the user to change his password. The security policy of an organization will generally define the amount of time between password changes. It is also important to set the minimum amount of time that a password must be used before it can be changed. This prevents users from changing their password when required to by the system, and then changing it right back to the old value.

```
[root@stationX]# chage [options] username
```

Common options used with the **chage** command:

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| <b>-m</b>      | minimum days between password changes                                 |
| <b>-M</b>      | maximum days between password changes                                 |
| <b>-I</b>      | number of days inactive since password expired before locking account |
| <b>-E date</b> | expire the account on this date (YYYY-MM-DD format)                   |
| <b>-W</b>      | number of days before a required change to start warnings             |
| <b>-l</b>      | list the settings                                                     |

**chage** *username* without options will let you change the settings interactively.



# Switching Accounts

- Syntax
  - **su [-] [user]**
  - **su [-] [user] -c *command***
- Allows the user to temporarily become another user
  - Default user is root
- The “-” option makes the new shell a login shell



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The **su** command is used to switch to another account from the command line. This command is most often used by system administrators to temporarily become the root user without logging out of their non-privileged account. It is very important to only use the root account when absolutely necessary, because of the awesome power of the root account. Day to day use of the system should never be conducted with the root account.

The password of the account being switched to must be supplied unless the superuser issued the **su** command.

Without the - option, the original user's environment is maintained. Using the - option causes the new shell to be a login shell which, among other things, unsets the original user's environment variables in the new shell. This is usually preferred so that actions performed as the new user will not have any effect on the old user, and so the new user's environment is available, rather than the old.

There are a number of options to the **su** command. See the documentation for more information.

Most systems log the use (or attempted use) of **su** to change to the root account. System administrators will often log on to the system as an ordinary user and then use **su** to gain access to the root account rather than log in directly as root so that the switch is logged.

Most systems administration tasks are best performed using **sudo**. **sudo** is safer than an **su** to root.

# sudo

- Users listed in `/etc/sudoers` execute commands with:
  - an effective user id of 0
  - group id of root's group
- An administrator will be contacted if a user not listed in `/etc/sudoers` attempts to use **sudo**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The purpose of **sudo** is to delegate root privileges to non-root users. It has many advantages over the **su** including not having to manage a shared password and logging of who actually executed commands, when and from where.

**sudo** access is controlled by `/etc/sudoers`. `/etc/sudoers` contains mappings of variables to reference groups of users, hosts, or commands. This file should be edited with **visudo**, which will open a copy of the current file with **vi** (or whatever editor is specified in the `EDITOR` environment variable) and check the syntax before implementing changes.

One common `sudoers` configuration is to grant full privileges to a user or group of trusted users with the expectation that they'll use the privileges responsibly and honestly. This simple configuration can be established with just one line in the `sudoers` file:

```
username ALL=(ALL) ALL
```

To give a specific group of users limited root privileges, edit the file with **visudo** as follows:

In the user alias specification section, list users and groups allowed to use the **sudo** command:

```
User_Alias LIMITEDTRUST=student1,student2
```

In the command alias specification section, list the commands specifically allowed or denied execution:

```
Cmnd_Alias MINIMUM=/etc/rc.d/init.d/httpd
```

In the user privilege specification section, list the users and groups allowed to use **sudo** and the commands that they may use:

```
LIMITEDTRUST ALL=MINIMUM
```

Users `student1` and `student2` can use **sudo** only with the commands listed with `MINIMUM` (**httpd**).

When delegating privileges to users that are not completely trusted, extreme caution must be employed. Many commands commonly used by root were not written with this sort of security in mind and are not appropriate for **sudo** access. Any command that launches an editor, pager or can be configured through environment variables is probably not safe. Shell scripts are also generally not safe unless they take care to reset the `PATH` environment variable.



# Network Users

- Information about users may be centrally stored and managed on a remote server
- Two types of information must always be provided for each user account
  - Account information: UID number, default shell, home directory, group memberships, and so on
  - Authentication: a way to tell that the password provided on login for an account is correct



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

We can store information about user and group accounts in local files like `/etc/passwd` on each workstation or server. However, it may be easier to keep account information synchronized among many computers by storing it centrally in a remote network server.

Two basic types of information must be provided for each user account. First, name service information must be provided to the standard C library, `glibc`. This information maps the account's username to a UID number, primary group (by GID number), GECOS field information (such as the user's real name), home directory, and default shell. This is controlled by a system called *Name Service Switch*, or NSS. Second, a mechanism must be provided which can determine if a password provided to authenticate login or other access to a particular account is the correct password for that account. This is configured for programs through the *Pluggable Authentication Modules* system, or PAM.

As shipped, NSS is configured to get information about users and groups only from local files. PAM is configured by default to authenticate users by encrypting the password provided to the login program and comparing it to the encrypted password information provided by NSS (and normally stored in `/etc/shadow`) for that username.



# Authentication Configuration

- **system-config-authentication**
  - GUI tool to configure authentication
  - For text-based tool, use **authconfig-tui**
  - Load **authconfig-gtk** RPM
- Supported account information services:
  - (local files), NIS, LDAP, Hesiod, Winbind
- Supported authentication mechanisms:
  - (NSS), Kerberos, LDAP, SmartCard, SMB, Winbind



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The mechanisms used to manage user information can be changed with **system-config-authentication**. By default, this provides a graphical tool to configure network authentication. The interface provides two tabs: **User Information** (which changes NSS settings) and **Authentication** (which changes PAM settings). Check boxes are used to select which user information sources or authentication mechanisms are desired, and then each service must be configured through button selections. In addition, use of shadow passwords or the MD5 password encryption algorithm can be toggled on or off. The **--nox** option will open this tool as a text-based menu interface instead, usable without an X display. In that case, the first screen will allow selection of the user information sources or authentication mechanisms, and subsequent screens will be used to configure those interfaces.

For **User Information** five data sources are supported:

If nothing is selected, only local files will be used as a source of NSS information. NIS gets information from database maps stored on a NIS server. LDAP allows account information to be stored as entries in a LDAP directory server. Hesiod stores user information as special resource records in a DNS name server, and its use is relatively uncommon. Winbind uses winbindd to automatically map accounts stored in a Microsoft Windows domain controller to Linux users by storing SID to UID/GID mappings in a database and automatically generating any other NSS information that is required.

For **Authentication**, six data sources are supported:

If nothing is selected, it is assumed that NSS will provide an encrypted password with the other NSS user information that can be compared to the entered password normally. Kerberos authenticates users by requesting a *ticket* for the user from the Kerberos server, and if the user's password decrypts the ticket the authentication passes. LDAP authentication maps the username provided to a LDAP directory entry and tries to bind to the directory using that entry and the provided password; if this succeeds, the authentication passes. SmartCard authentication allows a SmartCard to be used to login. Once removed, it can also be used to lock the system. SMB and Winbind use different approaches to authenticate using a Microsoft Windows® domain controller.

The older version of this tool in Red Hat Enterprise Linux 3 and earlier was **redhat-config-authentication**, and it had a slightly different look and feel. The **authconfig** command can be called as **authconfig-gtk** or **authconfig-tui** for graphical or text-based utility respectively, and also supports a **--kickstart** option which can make these settings through command-line flags.



## Example: NIS Configuration

- Must install `ypbind` and `portmap` RPMs
- Run **system-config-authentication**
  - Enable NIS to provide User Information
  - Specify NIS server and NIS domain name
  - Keep default authentication (through NSS)
- What does this actually do?
  - Five text-based configuration files are changed



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

One popular service used to centrally manage system and account information is NIS. NIS uses one or more NIS servers, each running `ypserv`, to share information with NIS client systems, running `ypbind`. The master server may also run `rpc.yppasswdd`, which allows users on NIS clients to update the passwords stored in NIS. Both NIS clients and NIS servers also must run a local service called `portmap` which helps remote systems contact the local `ypserv` or `ypbind` program. Clients and servers which communicate with each other are normally members of the same NIS domain, identified by an arbitrary name.

NIS servers typically are used to synchronize account information. They can share the contents of the `/etc/passwd`, `/etc/shadow`, `/etc/group`, and `/etc/gpasswd` files by converting them into *NIS maps*. Each NIS map consists of a set of key/value pairs. For instance, one typical NIS map used is `passwd.byname`, where the key is a username and the value is the matching line of user information for that account in `/etc/passwd` format.

The easiest way to set up a client to use an existing NIS server is to install the `portmap` and `ypbind` packages (which are part of the base install), and run **system-config-authentication**. Under **User Information**, enable NIS, and then a NIS domain and a NIS server for that domain must be specified. No changes are necessary under **Authentication** if NIS will be used for authentication, since it provides password hash information through NSS.

What does this change? The variable `NISDOMAIN` is set in `/etc/sysconfig/network` to the NIS domain's name, and the `domainname` command is run to set it. The `/etc/yp.conf` file has a line added to it which specifies which server to use for that NIS domain. The `/etc/nsswitch.conf` file is modified to specify that NIS should be used as a source of information for password, shadow, group, and other lookups. The `USENIS=yes` variable is set in `/etc/sysconfig/authconfig`. Finally, `/etc/pam.d/system-auth-ac` is modified so that password change requests for NIS accounts will be sent to the `rpc.yppasswdd` service running on the NIS master server.

NIS is a relatively insecure service. Kerberos authentication can be used in conjunction with NIS to improve password security. A better solution might use LDAP protected with TLS (SSL) encryption to store name service information in place of NIS. However, these solutions can be more complex to set up and manage.



## Example: LDAP Configuration

- Must install `nss-ldap` and `openldap` RPMs
- Run **system-config-authentication**
  - Enable LDAP to provide User Information
  - Specify server, the search base DN, and TLS
  - Enable LDAP to provide Authentication
- What does this actually do?
  - Five text-based configuration files are changed



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

LDAP is a protocol used to talk to a distributed directory service based on *X.500*, which can be used to store system and account information. Client programs can use the LDAP protocol to get information from directory servers running **slapd**, the standalone LDAP service. The OpenLDAP packages in the distribution provide a **slapd** server implementation as well as client tools and development libraries which can be used to work with LDAP services. Information is stored in the directory in individual entries organized into a hierarchical tree. Each entry and its location in the tree is uniquely identified by its *distinguished name*, or DN. A particular LDAP server is normally responsible for only the part of the tree under a particular entry, and the DN of this entry is normally used as a base DN for searches when looking up information stored in that server. It may help to think of this base DN like a DNS domain name, and the entire LDAP directory tree like all of DNS.

One use of an LDAP directory service is to synchronize account information between multiple networked systems. An individual entry may represent a single user (by using information from the relevant lines in `/etc/passwd` and `/etc/shadow`), or a single group (using information from the relevant line in `/etc/group`), or may store other information.

The easiest way to set up a client to use an existing LDAP server is to install the `nss_ldap` and `openldap` packages, and run **system-config-authentication**. To get NSS information from LDAP, under **User Information**, enable LDAP, and then specify an LDAP server, and a base DN to use for searches. If the LDAP service provides standard password hashes to NSS, this may be sufficient. Alternatively, under **Authentication** you can also enable LDAP, which will let PAM test and change passwords by accessing the directory service using the DN of the account's entry and the password entered. If you select LDAP on the **Authentication** tab, it is crucial to also check **Use TLS to encrypt connections**, or your unencrypted password will be transmitted over the network to the LDAP server as clear-text on every authentication!

What do these settings change? The `/etc/ldap.conf` file is modified to specify the location of your LDAP server, your search base DN, and whether or not TLS is enabled. The `/etc/openldap/ldap.conf` file is modified with the same information so that command-line OpenLDAP tools and the automounter can use the same server and base DN. The `/etc/nsswitch.conf` file is modified to specify that LDAP should be used as a source of information for password, shadow, group, and other lookups. The `USELDAPAUTH=yes` and/or `USELDAP=yes` variables are set in `/etc/sysconfig/authconfig`. Finally, `/etc/pam.d/system-auth-ac` is modified so that PAM will try accessing the directory service as your account entry to authenticate access.



# SUID and SGID Executables

- Normally processes started by a user run under the user and group security context of that user
- SUID and/or SGID bits set on an executable file cause it to run under the user and/or group security context of the file's owner and/or group



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

When a user starts a process, it runs with the permissions of that user. For example, if you run `vi`, and try to edit a file which you do not have permission to read or write, the operation will fail. However, if the SUID or SGID bit is set on an executable, it runs with the permissions of its owner (or group owner). For example, consider the file `/etc/shadow` that stores users' encrypted passwords:

```
-r----- 1 root root 805 Sep 29 11:19 /etc/shadow
```

The file is owned by root, who has exclusive read access. Users may still change their passwords with the `passwd` command, because the `passwd` command has its SUID bit set, and is owned by root:

```
-rwsr-xr-x 1 root root 13536 Jul 12 05:56 /usr/bin/passwd
```

SUID and SGID bits are set using the `chmod` command:

SUID:

```
[root@stationX]# chmod u+s filename
```

SGID:

```
[root@stationX]# chmod g+s filename
```

Since the SUID and SGID permissions are displayed overlying the execute permission for either user or group, respectively, the case of the permission indicates whether the execute permission is turned on or off. For example, if a capital is in the execute field, the SUID or SGID permission is on and the execute permission is off. If a lowercase is in the execute field, both the SUID or SGID and the execute permissions are on.

For security reasons, SUID and SGID permissions are not honored when set on non-compiled programs, such as shell scripts.

*So -s = execute + SUID  
So -s only SUID has been granted*

# SGID Directories

- Used to create a collaborative directory
- Normally, files created in a directory belong to the user's default group
- When a file is created in a directory with the SGID bit set, it belongs to the same group as the directory



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

When a file is created in a directory, it belongs to the primary group of the user that created the file.

```
[root@stationX]# mkdir /shared
[root@stationX]# ls -ld /shared
drwxr-xr-x 2 root root 4096 Feb 27 11:28 /shared
[root@stationX]## touch /shared/first
[root@stationX]## ls -l /shared
-rw-r--r-- 1 root root 0 Feb 27 11:33 first
```

However, if the *setgid* (SGID or Set Group ID) bit is set for the directory, new files that are created in the directory have their group ownership set to the same group ownership as the directory. This provides a mechanism to allow one level of access to users, who are members of the same group that owns the directory while allowing a different level of access to non-member users of files in the directory.

Recall that permissions can be set with **chmod**:

```
[root@stationX]# chmod g+s directory
```

This sets the SGID bit without affecting current permissions. Alternately:

```
[root@stationX]# chmod 2770 directory
```

The above syntax sets the SGID bit and gives read, write, and execute permissions to the owner of the directory and members of the group whose ownership is on that directory.

```
[root@stationX]# chgrp mail /shared
[root@stationX]# ls -ld /shared
drwxr-xr-x 2 root mail 4096 Feb 27 11:28 /shared
[root@stationX]# chmod g+s /shared
[root@stationX]# ls -ld /shared
drwxr-sr-x 2 root mail 4096 Feb 27 11:28 /shared
```

In this example, all future files will be created as members of the mail group.

```
[root@stationX]# touch /shared/second
[root@stationX]# ls -l /shared
-rw-r--r-- 1 root root 0 Feb 27 11:33 first
-rw-r--r-- 1 root mail 0 Feb 27 11:40 second
```



## The Sticky Bit

- Normally users with write permissions to a directory can delete any file in that directory regardless of that file's permissions or ownership
- With the sticky bit set on a directory, only the owner of a file can delete the file
- Example:

```
ls -ld /tmp
drwxrwxrwt 12 root root 4096 Nov 2 15:44 /tmp
 ^
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

It may seem counterintuitive that write permissions on a directory would allow one user to delete another user's file within that directory. Consider, however, that a directory is really just a file itself whose contents are references to other files. Deleting a file is therefore an edit to a directory file's list of other files.

Many users need to be able to create and delete files in /tmp. Even if users do not actively create files in /tmp, many applications they run will use /tmp as a location for temporary files. Setting the sticky bit prevents users from deleting each others' files, even though they have full access to the directory.

Note that the sticky bit on /tmp is set by default, and can be seen as a "t" in the file permissions:

```
drwxrwxrwt 13 root root 4096 Sep 29 12:42 /tmp
```

To set the sticky bit on a directory, use **chmod**:

```
[root@stationX]# chmod o+t /home/share
```

The sticky bit will appear as a **T** if the directory's execute permission for "others" is off.



7/10

## Default File Permissions

- Read and write (not execute) for all is the default for files
- Read, write and execute is the default for directories
- **umask** can be used to withhold permissions on file creation
- Users' umask is 022
  - Files will have permissions of 644
  - Directories will have permissions of 755
  - May need to change to 002 for group collaboration

us



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Without a umask in effect, any file created will have 666 permissions. This means that anyone on the system will have read and write access to any newly-created file. In order to withhold permissions we use a umask. The **umask** lists the permissions to withhold. A umask of 002 will result in files created with 664 permissions.

The default umask on a Red Hat Enterprise Linux system is 022. To change your umask to 002, use the **umask** command:

```
[student@stationX]# umask 002
```

Note that when setting your umask from the command line, the next time you login your umask will be set back to your default, so this is typically set by one of the bash initialization scripts, such as at the bottom of `~/ .bashrc`.

# Access Control Lists (ACLs)

- Grant rwx access to files and directories for multiple users or groups
  - `mount -o acl /directory`
  - `getfacl file|directory`
  - `setfacl -m u:gandolf:rwx file|directory`
  - `setfacl -m g:nazgul:rw file|directory`
  - `setfacl -m d:u:frodo:rw directory`
  - `setfacl -x u:samwise file|directory`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The ext3 filesystem includes support for access control lists which allow finer grained control of file system permissions than are possible with the standard three access categories that are normally provided. Many filesystem commands, such as **cp** and **mv**, have been modified to copy the associated ACL's for a file.

In order to enable ACLs on a file system, the file system must be mounted with the **acl mount** option. Filesystems created during installation include the **acl** flag in their default mount option. To remount the `/home/` directory with the **acl** option, run the following:

```
[root@stationX]# mount -o remount,acl /home/
```

To view the ACL's for a file, use the **getfacl** command:

```
[root@stationX]# getfacl /home/schedule.txt
getfacl: Removing leading '/' from absolute path names
file: home/schedule.txt
owner: root
group: root
user::rw-
user:bob:rwx
group::r--
group:admins:rw
mask::rwx
other::r--
```

ACL's can be modified using the **setfacl** command:

```
[root@stationX]# setfacl -m u:visitor:rx /home/schedule.txt
```

The above command would grant the user visitor read and execute access to the file `/home/schedule.txt`.

To remove (expunge) an ACL:

```
[root@stationX]# setfacl -x u:visitor /home/schedule.txt
```

On directories, default access control lists can be used. If we wanted all newly created content of a directory to be readable and writable by the user student:

```
[root@stationX]# setfacl -m d:u:student:rw /home/share/project/
```



# SELinux

- Mandatory Access Control (MAC) -vs- Discretionary Access Control (DAC)
- A rule set called the *policy* determines how strict the control
- Processes are either restricted or unconfined
- The policy defines what resources restricted processes are allowed to access
- Any action that is not explicitly allowed is, by default, denied



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

In an effort to deal with ever increasing threats to their data systems, the US government assigned the National Security Agency (NSA) with the task of developing a single set of rules that all other agencies would follow in handling confidential information. By evaluating previous breaches, the NSA determined that a major hurdle to securing data was internal users bypassing local security. In some cases, users would inadvertently open access to a system. A classic example would be a user executing the command **chmod 777 ~**. To the user, this may seem an easy way to allow co-workers to share files. They may not realize that this gives everyone in the world access to potentially confidential information.

In more extreme cases, users would intentionally disable security for more insidious reasons. Both of these situations are examples of a user having the discretion to control the access to their system. The NSA felt the solution was to have systems implement *Mandatory Access Control* (MAC) over the users. In MAC, a set of rules, known as the *policy*, identify what a process is allowed to do. Anything that is not explicitly permitted is, by default, denied. Ideally, different policies could be implemented depending upon how strict the security needs.

The NSA's first implementation of MAC was a system called Mach, which introduced the concept of *Type Enforcement* (TE). Objects (files, directories, resources) were assigned a type value. Users and processes were also assigned a type value. The policy contains rules that allow a user or process to access objects. It was possible for a policy to be so *strict*, it was difficult to manage, so occasionally users and processes were allowed to be *unconfined*. This meant the policy did not apply to that user or process.

The NSA decided Mach made a better security framework than operating system. To get this framework deployed on production systems, the NSA needed access to the OS source code. They took advantage of the open source Linux kernel, and developed a set of patches to enable MAC. These patches became known as Security Enhanced Linux, or SELinux for short. They were later refined and incorporated into the kernel source by Red Hat.

## SELinux, continued

- All files and processes have a *security context*
- The context has several elements, depending on the security needs
  - user:role:type:sensitivity:category
  - user\_u:object\_r:tmp\_t:s0:c0
  - Not all systems will display s0:c0
- **ls -Z**
- **ps -Z**
  - Usually paired with other options, such as **-e**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

There is an old UNIX saying that “everything is a file”. Traditional file access is controlled by user, group, and permission settings. To SELinux, everything is an object and access is controlled by security elements stored in the the inode's extended attribute fields. Collectively, the elements are called the security context. Presently, there are five supported elements, although all five may not be present on all systems:

|             |                                                                                                                                                                                                                                                                                                                                                                           |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| user        | Indicates the type of user that is logged into the system. If a user logs in as root, they will have the user value of <code>root</code> . Other users will have a value of <code>user_u</code> . If they escalate their privileges with <code>su</code> , they will still have the user value of <code>user_u</code> . Processes have a value of <code>system_u</code> . |
| role        | Defines the purpose of the particular file, process, or user. Files have the role of <code>object_r</code> . Processes get the role of <code>system_r</code> . Users also have the role of <code>system_r</code> , because (to Linux) users are similar to processes.                                                                                                     |
| type        | Used by Type Enforcement to specify the nature of the data in a file or processes. Rules within the policy say what process types can access which file types.                                                                                                                                                                                                            |
| sensitivity | A security classification sometimes used by government agencies.                                                                                                                                                                                                                                                                                                          |
| category    | Similar to group, but can block root's access to confidential data.                                                                                                                                                                                                                                                                                                       |

To view the security context of a file, use the `ls` command's **-Z** option:

```
[root@shadex4 ~]# ls -Z /root/anaconda-ks.cfg /var/log/messages
-rw-r--r-- root root user_u:object_r:user_home_t anaconda-ks.cfg
-rw----- root root system_u:object_r:var_log_t /var/log/messages
```

Generally speaking, files inherit a directory's security context:

```
[root@stationX ~]# ls -Zd /etc /etc/hosts
drwxr-xr-x root root system_u:object_r:etc_t /etc
-rw-r--r-- root root system_u:object_r:etc_t /etc/hosts
```

Some files, however, get a unique security context, for added security:

```
[root@stationX ~]# ls -Z /etc/shadow /etc/aliases
-r----- root root system_u:object_r:shadow_t /etc/shadow
```



```
rw-r--r-- root root system_u:object_r:etc_aliases_t /etc/aliases
```

If a system were running under the most secure configuration, everything would be restricted by SELinux. This is not practical in most cases. For this reason, SELinux is being deployed in phases. In Red Hat Enterprise Linux 4, SELinux was protecting 13 processes. In the initial release of Red Hat Enterprise Linux 5, this count had increased to 88.

To determine if a process is protected, use the **ps** command's **-Z** option:

```
[root@stationX ~]# ps -ZC syslogd,bash
LABEL PID TTY TIME CMD
system_u:system_r:syslogd_t 1888 ? 00:00:00 syslogd
user_u:system_r:unconfined_t 2583 pts/0 00:00:00 bash
```

Any process whose type is *unconfined\_t*, is not yet restricted by SELinux. To view the entire process stack, use either **ps -eZ** or **ps Zax**.

*Note:* A restricted process is sometimes called *protected*, though it is the data that is protected, not the process.



# SELinux: Targeted Policy

- The targeted policy is loaded at install time
- Most local processes are *unconfined*
- Principally uses the type element for *type enforcement*
- The security context can be changed with **chcon**
  - **chcon -t tmp\_t /etc/hosts**
- Safer to use **restorecon**
  - **restorecon /etc/hosts**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The source code that allows SELinux to protect a process is integrated into the kernel, but the rules that define how SELinux enforces security is defined in the policy. The NSA's most secure policy is called the *strict* policy. By default, Red Hat uses the *targeted* policy. The targeted policy "targets" certain processes to be restricted, and then enforces their access to files and resources. Other policies also exist, such as the *Multi Level Security* (MLS). All policies could be thought of as a subset of the Strict policy.

The policy defines the elements that can be used, and whether users are allowed to manipulate the elements. Since the targeted policy uses Type Enforcement as its principal security mechanism, it pays the most attention to the type element of the security context. The policy allows the **chcon** command to change the security context.

```
[root@stationX ~]# ls -Z install.log
-rw-r--r-- root root root:object_r:user_home_t install.log
[root@stationX ~]# chcon -t etc_t install.log
[root@stationX ~]# ls -Z install.log
-rw-r--r-- root root root:object_r:etc_t install.log
```

The **chcon** command can only use types that are defined in the policy. Rather than memorizing all the possible types that can be used, **chcon --reference** can take the security context from one object, and apply it to another.

```
[root@stationX ~]# ls -Z anaconda-ks.cfg
-rw----- root root system_u:object_r:user_home_t anaconda-ks.cfg
[root@stationX ~]# chcon --reference /etc/shadow anaconda-ks.cfg
[root@stationX ~]# ls -Z anaconda-ks.cfg
-rw----- root root system_u:object_r:shadow_t anaconda-ks.cfg
```

A safer alternative is the **restorecon** command. With **restorecon**, the policy determines and applies the object's default context.

```
[root@stationX ~]# restorecon /root/*
[root@stationX ~]# ls -Z /root
-rw----- root root root:object_r:user_home_t anaconda-ks.cfg
-rw-r--r-- root root root:object_r:user_home_t install.log
```

# SELinux: Management

- Modes: Enforcing, Permissive, Disabled
  - Changing enforcement is allowed in the Targeted policy
  - **getenforce** *not persistent*
  - **setenforce 0 | 1**
  - Disable from GRUB with **selinux=0**
- `/etc/sysconfig/selinux`
- **system-config-securitylevel**
  - Change mode, Disabling requires reboot
- **system-config-selinux**
  - Booleans
- **setroubleshootd**
  - Advises on how to avoid errors, not ensure security!



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The policy for a system is defined in the `/etc/sysconfig/selinux` file as is the mode of operation. The policy can be Disabled, Enforcing, or Permissive. Disabled means the policy is ignored. Permissive is a mode for troubleshooting or development that logs policy violations, but does not prevent programs from running. Enforcing is the default mode.

The **getenforce** command can be used determine the system's current mode. The targeted policy allows the use of the **setenforce** command to toggle between the Enforcing (1) and Permissive (0) mode:

```
[root@stationX ~]# getenforce
Enforcing
[root@stationX ~]# setenforce 0
[root@stationX ~]# getenforce
Permissive
```

The only way to disable SELinux is to change `/etc/sysconfig/selinux`, and reboot, or use the **selinux=0** Grub kernel option.

The GUI tool, **system-config-selinux**, allows for changes of a few other SELinux options. The targeted policy allows certain features to be controlled through a set of *booleans*. These are predefined functions that can selectively turn off enforcement of certain daemons. It would be preferable to ensure objects have the correct security context rather than shutting off security features.

When an application attempts something that is not authorized by the policy, SELinux blocks access and an error is logged to `/var/log/messages`. The application is often unaware of why it failed. This can make troubleshooting difficult. To help in the troubleshooting process, the **setroubleshootd** daemon will alert you of the error by placing a warning icon on the alert panel. Clicking on the icon will display a possible fix for the error. It is important to realize that the proposed solution may not be the best solution for the problem.



## End of Unit 5

- Questions and Answers
- Summary
  - User and Group accounts
  - File ownership and permissions
  - Extended file modes: SUID / SGID / Sticky
  - Switching accounts with su
  - **umask** and the UPG scheme
  - ~~Shell environment~~
  - Setup NIS and LDAP
  - Use ACLs
  - Configure and troubleshoot SELinux



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.











# Lab 5

## User and Group Administration

---

- Goal:** To build skills for user and group administration.
- Estimated Duration:** 45 Minutes
- Lab Setup:** Instructor: Start **ypserv**.
- Situation:** You need to setup groups and accounts for several users in your company. Each department also requires a shared directory for their files.

## Sequence 1: Creating the groups and users

**Scenario:** You need to set up groups for different departments in your company. You also need to set up user accounts for the employees in those departments.

**Deliverable:** A system with users joshua and alex in the sales group; dax and bryan in the hr group; zak and ed in the web group, and manager in the sales, hr, and web groups.

### Instructions:

1. Make certain that all newly created users will create group-writable files.
2. Add accounts for the following seven users to your system: joshua, alex, dax, bryan, zak, ed, and manager. Assign each user this password: `password`
3. Add the following groups to the system:
  - sales (GID:200)
  - hr (GID: 201)
  - web (GID: 202)
4. Why should you set the GID in this manner instead of allowing the system to set the GID by default?  

---

*they'll own all the files*
5. Add joshua and alex to the sales auxiliary group, dax and bryan to the hr auxiliary group. Add zak and ed to the web auxiliary group. Add manager to all of these auxiliary groups.
6. How can you add the web group to dax's auxiliary groups?  

---

*usermod -a -G web dax*
7. You can login as each user and use the `id` command to verify that they are in the appropriate groups. How else might you verify this information?  

---

*ls /etc/passwd file*

## Sequence 2: Setting up shared directories

**Scenario:** Each department for which you created a group also needs a shared directory. This will allow users in each department to share files, but will prevent users in other departments from altering, or even seeing those files.

**Deliverable:** A shared directory for each department that allows only users in that department to enter it or create, view, and alter files within.

### Instructions:

1. Create a directory called `/depts` with `sales`, `hr`, and `web` subdirectories. ✓
2. Set the group ownership of each directory to the group with the matching name. ✓
3. Check the permissions on the `/depts` directory to verify that everyone can access, but not write to the directory. Change the permissions on each each subdirectory to grant full permissions (`rwX`) to the group and deny access to others. ✓
4. Files created within those directories should be owned by the appropriate group. Set the appropriate permissions. ✓
5. Experiment by logging in as each user and creating or altering files in each of the directories. Only manager should be able to enter all the directories. ✓



## Sequence 3: Access Control Lists

- Scenario:** Some data needs to be accessible to several groups. Use ACLs to accomplish this.
- Deliverable:** Directories accessible by several groups and users.
- System Setup:** *Note:* The `/depts/` directory is part of the `/` partition. Since this partition was created during installation it automatically includes the `acl` mount option. Filesystems created after installation must be specifically mounted with the `acl` option. See the “Filesystem Management” unit for details.

### Instructions:

1. ✓ Create a new directory in `/depts/` called `tech`. Change the permissions such that root is the owner and `hr` is the group.
2. ✓ Use ACLs to give full permission for `/depts/tech/` to the `web` group. ✓
3. ✓ Allow alex read/execute (but not write) permission on the `/depts/tech/` directory. Set a default ACL of read/write for alex on that directory.
4. ✓ Create some files in `/depts/tech/` as several of the users and verify access. Does alex or joshua have access to files? Does manager?

## Sequence 4: Understanding Security Context

**Scenario:** A set of files need to be migrated to the webserver and made to operate within the SELinux framework.

**Deliverable:** A webserver with three viewable pages.

### Instructions:

1. ✓ Install the **httpd** package and test that the webserver is displaying pages. ✓
2. ✓ Ensure SELinux is in the *Enforcing* mode. ✓
3. ✓ Create a user account named `student` on your system. ✓
4. ✓ In student's home directory create a file named `home.html` containing a line that reads "*home directory*". Also, create a file in `/tmp` named `tmp.html` containing a line that reads "*temporary directory*". Inspect each file's security context.
5. ✓ Standard Apache webserver configuration does not allow access to the home or temporary directories. Move `home.html` to the server's default directory: `/var/www/html`. Investigate the file's security context.
6. ✓ Attempt to view the webpage with the command:  
✓  
**# elinks -dump http://localhost/home.html**  
✓ You should be forbidden from viewing the page. Change SELinux to the *Permissive* mode and try again.
7. ✓ Return SELinux to the *Enforcing* mode. Copy `tmp.html` to `/var/www/html` and investigate the security context. Attempt to view the webpage.
8. ✓ Use **restorecon** to correct the context of the `home.html` file. Use **elinks** to test your fix.

## Sequence 5: Client-side NIS account management

**Scenario:** Your site is centrally managing user account information in a NIS directory service on 192.168.0.254, using the NIS domain name `notexample`. Set up your client to get user information from NIS, authenticating users with the NSS information provided by the NIS server.

*Note:* Home directories for these users are exported through NFS to all workstations as well. You can set up the automounter to automatically mount and unmount these directories as needed. Automounter is covered in the Filesystem Management unit.

**Deliverable:** Your system configured as an NIS client.

### Instructions:

1. ✓ Install the necessary RPMs on your system if they are not already installed.
2. Configure your client to use NIS for authentication. Use `notexample` as the domain and `192.168.0.254` as the server.
3. Restart the **sshd** service to make sure that it registers the changes to authentication. You can test NIS with the user `guest20XX` (where `XX` is a two-digit version of your station number), and a password of `password`. **ssh** should succeed, but display an error message about the missing home directory.

2013  
password



## Sequence 6: Client-side LDAP account management

**Scenario:** Your site is migrating new user account information into an LDAP directory service, also served by `server1.example.com` under the search base `dc=example,dc=com`. The administrator of the LDAP server requires clients to use TLS (SSL) encryption, and the LDAP server's TLS certificate is digitally signed by a locally-run TLS Certificate Authority so that clients may verify that they are communicating with the real LDAP server.

Using TLS encryption, set up your client system to get user information and authenticate users using the LDAP server. Home directories for these LDAP-managed users use the same NFS export as your old NIS-managed users, so you will not have to make any changes to your automounter configuration.

**Deliverable:** Your system configured as an LDAP client.

**Instructions:**

*must get server1.example.com*

1.  Install the necessary RPMs on your system. Your system should already have the `authconfig` and `authconfig-gtk`. Several RPMs are needed for LDAP: `openldap`, `openldap-clients`, and `nss_ldap`.
2. Use **system-config-authentication** to configure your client to use LDAP for user information and authentication. The LDAP server is `server1.example.com`. The Search Base DN is: `dc=example,dc=com`. The server is only available via TLS. You will need to *Download CA Certificate* from the following URL:  
  
`ftp://server1/pub/EXAMPLE-CA-CERT`
3. Restart the **sshd** service to make sure that it registers the changes to authentication. You can use the user account `ldapuserX` with a password of `password` for testing.
4. If you see any errors, look at the logs in `/var/log/messages` and `/var/log/secure`. You can also try running the command **ldapsearch -x -Z**, which should dump user information in LDIF format from your LDAP server to standard output if the server is reachable.

TLS can be tested with **openssl s\_client**. See `s_client(1)` for details.

## Sequence 1 Solutions

1. Make certain that all newly created users will create group-writable files.
  - a. Add a umask entry to /etc/skel/.bashrc making group-writable files (002).

```
[root@stationX]# echo 'umask 002' >> /etc/skel/.bashrc
```

2. Add accounts for the following seven users to your system: joshua, alex, dax, bryan, zak, ed, and manager.

Since you need to add several users, a **for**-loop may speed things up. This can be entered on the command line or placed in a file and run as a shell script.

```
for USER in joshua alex dax bryan zak ed manager
do
 useradd $USER
 echo password | passwd --stdin $USER
done
```

Note that this sets a password of password for each user.

3. Add the following groups to the system:

- sales (GID:200)
- hr (GID: 201)
- web (GID: 202)

```
a. [root@stationX]# groupadd -g 200 sales
```

```
b. [root@stationX]# groupadd -g 201 hr
```

```
c. [root@stationX]# groupadd -g 202 web
```

4. Why should you set the GID in this manner instead of allowing the system to set the GID by default?

The primary group for new users will be created with lowest available GID above 500. Most administrators want to keep additional groups in a specific range of GIDs to ease management.

5. Add joshua and alex to the sales auxiliary group, dax and bryan to the hr auxiliary group. Add zak and ed to the web auxiliary group. Add manager to all auxiliary groups.

You can use **usermod -G** to do this:

```
[root@stationX]# usermod -G sales joshua
[root@stationX]# usermod -G sales alex
[root@stationX]# usermod -G hr dax
```



```
[root@stationX]# usermod -G hr bryan
[root@stationX]# usermod -G web zak
[root@stationX]# usermod -G web ed
[root@stationX]# usermod -G sales,hr,web manager
```

6. How can you add the web group to dax's auxiliary groups?

```
[root@stationX]# usermod -G web,hr dax
```

-OR-

```
[root@stationX]# usermod -a -G web dax
```

If you had run the command **usermod -G web dax**, that would have removed dax from the hr group and any other auxiliary groups to which dax may have belonged.

7. You can login as each user and use the **id** command to verify that they are in the appropriate groups. How else might you verify this information?

You can use **su - user** and run the **id** command, or simply use the username as an argument to **id**:

```
for USER in joshua alex dax bryan zak ed manager
do
id $USER
done
```



## Sequence 2 Solutions

1. Create a directory called /depts with sales, hr, and web subdirectories.  

```
[root@stationX]# mkdir -p /depts/{sales,hr,web}
```
2. Set the group ownership of each directory to the group with the matching name.  

```
[root@stationX]# for GROUP in sales hr web
do
chgrp $GROUP /depts/$GROUP
done
```
3. Check the permissions on the /depts directory to verify that everyone can access, but not write to the directory. Change the permissions on each each subdirectory to grant full permissions (rwx) to the group and deny access to others.
  - a. 

```
[root@stationX]# ls -ld /depts
drwxr-xr-x 5 root root 4096 May 4 06:04 /depts
```
  - b. 

```
[root@stationX]# chmod g=rwx,o= /depts/*
```

-OR-

```
[root@stationX]# chmod 770 /depts/*
```
4. Files created within those directories should be owned by the appropriate group. Set the appropriate permissions.  

```
[root@stationX]# chmod g+s /depts/*
```
5. Experiment by logging in as each user and creating or altering files in each of the directories. Only manager should be able to enter all the directories.

The easiest way is probably **su -**, but make sure to include the dash and to exit one **su** session before starting another.

```
[root@stationX]# su - joshua
[joshua@stationX]$ touch /depts/sales/test
[joshua@stationX]$ exit
[root@stationX]# su - alex
[alex@stationX]$ touch /depts/sales/test
[alex@stationX]$ exit
[root@stationX]# su - dax
[dax@stationX]$ touch /depts/hr/test
[dax@stationX]$ exit
[root@stationX]# su - bryan
[bryan@stationX]$ touch /depts/hr/test
[bryan@stationX]$ exit
[root@stationX]# su - zak
```

```
[zak@stationX]$ touch /depts/web/test
[zak@stationX]$ exit
[root@stationX]# su - ed
[ed@stationX]$ touch /depts/web/test
[ed@stationX]$ exit
[root@stationX]# su - manager
[manager@stationX]$ touch /depts/sales/test
[manager@stationX]$ touch /depts/hr/test
[manager@stationX]$ touch /depts/web/test
[manager@stationX]$ exit
```



## Sequence 3 Solutions

1. Create a new directory in /depts/ called tech. Change the permissions such that root is the owner and hr is the group. Give full access to the user and group, and no access to anyone else. Don't forget to set the SGID bit.
  - a. `[root@stationX]# mkdir /depts/tech/`
  - b. `[root@stationX]# chown root:hr /depts/tech/`
  - c. `[root@stationX]# chmod 2770 /depts/tech/`
2. Use ACLs to give full permission for /depts/tech/ to the web group.
  - a. `setfacl -m g:web:rwX /depts/tech/`
3. Allow alex read/execute (but not write) permission on the /depts/tech/ directory. Set a default ACL of read/write for alex on that directory.
  - a. `[root@stationX]# setfacl -m u:alex:rx /depts/tech/`
  - b. `[root@stationX]# setfacl -m d:u:alex:rw /depts/tech/`
4. Create some files in /depts/tech/ as several of the users and verify access. Does alex or joshua have access to files? Does manager?
  - a. 

```
[root@stationX]# su - joshua
[joshua@stationX]$ touch /depts/tech/joshua
touch: cannot touch `/depts/tech/joshua': Permission denied
[joshua@stationX]$ exit

[root@stationX]# su - manager
[manager@stationX]$ touch /depts/tech/manager
[manager@stationX]$ getfacl /depts/tech/manager
getfacl: Removing leading '/' from absolute path names
file: depts/tech/manager
owner: manager
group: hr
user::rw-
user:alex:rw-
group::rwx
mask::rw-
other::---
[manager@stationX]$ exit

[root@stationX]# su - alex
[alex@stationX]$ touch /depts/tech/manager
[alex@stationX]$ touch /depts/tech/alex
touch: cannot touch `/depts/tech/alex': Permission denied
```



```
[alex@stationX]$ exit
```

## Sequence 4 Solutions

1. Install the **httpd** package and test that the webserver is displaying pages.

Use **yum** to install **httpd**.

```
yum install httpd -y
... output truncated ...
```

Start the service:

```
chkconfig httpd on
service httpd start
Starting httpd: [OK]
```

You could test the server with Firefox, but **elinks** might be faster.

```
elinks -dump http://localhost | head
 Red Hat Enterprise Linux Test Page
... output truncated ...
```

2. Ensure SELinux is in the *Enforcing* mode.

```
setenforce 1
getenforce
Enforcing
```

3. Create a user account named **student** on your system.

```
useradd student
```

4. In **student**'s home directory create a file named **home.html** containing a line that reads "*home directory*". Also, create a file in **/tmp** named **tmp.html** containing a line that reads "*temporary directory*".

```
echo "home directory" > /home/student/home.html
echo "temporary directory" > /tmp/tmp.html
```

Inspect each file's security context.

```
ls -Z /home/student/home.html
-rw-r--r-- root root user_u:object_r:user_home_t home.html
ls -Z /tmp/tmp.html
-rw-r--r-- root root user_u:object_r:tmp_t tmp.html
```

Notice that **home.html** and **tmp.html** have different type elements in their security context.

5. Standard Apache webserver configuration does not allow access to the home or temporary directories. Move **home.html** to the server's default directory: **/var/www/html**. Investigate the file's security context.

```
mv /home/student/home.html /var/www/html
ls -Z /var/www/html
-rw-r--r-- root root user_u:object_r:user_home_t home.html
```

Notice that security context information was maintained during the move.

6. Attempt to view the webpage with the command:

```
elinks -dump http://localhost/home.html
Forbidden
... output truncated ...
```

Change SELinux to the *Permissive* mode and try again.

```
setenforce 0
elinks -dump http://localhost/home.html
home directory
```

7. Return SELinux to the *Enforcing* mode.

```
setenforce 1
```

Copy tmp.html to /var/www/html and investigate the security context. Attempt to view the webpage.

```
cp /tmp/tmp.html /var/www/html
ls -Z /var/www/html
-rw-r--r-- root root user_u:object_r:user_home_t home.html ✓
-rw-r--r-- root root user_u:object_r:httpd_sys_content_t tmp.html ✓
elinks -dump http://localhost/tmp.html
temporary directory
```

Since /var/www/html/tmp.html is a new file (a copy of the original) it inherited the security context of the parent directory. Since it has the correct context, the httpd process can view the file.

8. Use **restorecon** to correct the context of the home.html file. Use **elinks** to test your fix.

```
restorecon /var/www/html/home.html
ls -Z /var/www/html
-rw-r--r-- root root system_u:object_r:httpd_sys_content_t home.html ✓
-rw-r--r-- root root user_u:object_r:httpd_sys_content_t tmp.html ✓
elinks -dump http://localhost/home.html
home directory
```



## Sequence 5 Solutions

1. Install the necessary RPMs on your system if they are not already installed. For example, to test for portmap, you can run **rpm -q portmap**.

The RPMS are:

- portmap
- ypbind
- yp-tools
- authconfig
- authconfig-gtk

2. Configure your client to use NIS for authentication. Use `notexample` as the domain and `192.168.0.254` as the server.

If you are using the graphical interface, check the option **Enable NIS Support** on the **User Information** tab. You can leave the **Authentication** tab alone. Click the **Configure NIS...** button, and in the **NIS Settings** window specify `notexample` for the **NIS Domain** and `server1.example.com` for the **NIS Server**. Click on the **OK** button for the **NIS Settings** window, and again on the main window.

If you are using a (text-based) virtual console or the **authconfig-tui** command, then under **User Information** check **Use NIS**. Under **Authentication** leave **Use MD5 Passwords** and **Use Shadow Passwords** checked. Then select the **Next** button, and on the next screen specify `notexample` for **Domain** and `192.168.0.254` for **Server**. Select **Ok**.

Either way, `ypbind` should now start up successfully. If it does not, check the configuration for misspellings, or `/var/log/messages` for errors.

3. Restart the **sshd** service to make sure that it registers the changes to authentication. You can test NIS with the user `guest20XX` (where `XX` is a two-digit version of your station number), and a password of `password`. **ssh** should succeed, but display an error message about the missing home directory.

a. `[root@stationX]# service sshd restart`

- b. Try to ssh into your system as `guest20XX`. The password for this account is `password`.

`[root@stationX]# ssh guest20XX@stationX.example.com`

- c. The **ssh** command should succeed, but display an error message about the missing home directory. This could be fixed by using `automounter`.

## Sequence 6 Solutions

1. Install the necessary RPMs on your system. Your system should already have the `authconfig` and `authconfig-gtk`. Several RPMs are needed for LDAP: `openldap`, `openldap-clients`, and `nss_ldap`.

```
yum install -y openldap openldap-clients nss_ldap
```

2. Use **system-config-authentication** to configure your client to use LDAP for user information and authentication. In the graphical interface, on the **User Information** tab check **Enable LDAP Support**. Switch to the **Authentication** tab and also check **Enable LDAP Support**.

Select the **Configure LDAP...** button on either tab. On the window that opens, set your **LDAP Search Base DN** to `dc=example,dc=com`. Set your **LDAP Server** to `server1.example.com`.

Using TLS is critical for security if you enable LDAP on the Authentication tab, because the PAM module used for LDAP will send passwords as cleartext over the network to the directory server on each authentication if this is not selected. Finally, press the **Download CA Certificate** button and enter the following in the **Certificate URL**:

```
ftp://server1/pub/EXAMPLE-CA-CERT
```

*Note:* If you do not add the certificate, you will notice the warning box about the missing certificate file.

3. Restart the **sshd** service to make sure that it registers the changes to authentication. You can use the user account `ldapuserX` with a password of `password` for testing.

```
service sshd restart
```

```
ssh ldapuserX@stationX.example.com
```

4. If you see any errors, look at the logs in `/var/log/messages` and `/var/log/secure`. You can also try running the command **ldapsearch -x -Z**, which should dump user information in LDIF format from your LDAP server to standard output if the server is reachable.

TLS can be tested with **openssl s\_client**. See `s_client(1)` for details.

- a. `[root@stationX]# openssl s_client -connect server1.example.com:443`











## Unit 6

# Filesystem Management



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



# Objectives

Upon completion of this unit, you should be able to:

- Understand filesystem hierarchy
- Manage virtual memory
- Add new drives and partitions
- Mount NFS filesystems



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Overview: Adding New Filesystems to the Filesystem Tree

- Identify Device
- Partition Device
- Make Filesystem
- Label Filesystem
- Create entry in `/etc/fstab`
- Mount New Filesystem



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Identify Device

The first step in adding a new filesystems to the filesystem tree is to identify the device to be used. This is usually something like `/dev/hda` or `/dev/sda`.

## Partition Device

You may either use an entire device or a partition on a device. If partitioning is preferred then use a tool like **fdisk** to partition the device and choose a partition type. If the whole device is to be used as a single unit then move on to the next step.

## Make Filesystem

The filesystem is what organizes files on a device. The filesystem type used on Red Hat Enterprise Linux is `ext3`. There are a number of customizable options for filesystems, including block size, byte/inode ratio and journaling.

## Label Filesystem

While the filesystem label is a property of the filesystem, it makes sense to see it as a separate logical step. Filesystem labeling creates a layer of abstraction that allows you to identify filesystems by a unique label rather than the device name. This allows device names to change without effecting system boot. This step is not essential, but it is highly recommended.

## Create entry in `/etc/fstab`

The `/etc/fstab` is what `/etc/rc.sysinit` uses to mount all filesystems at startup. It is essential that all entries in `/etc/fstab` are correct or the system will not boot properly.

## Mount the New Filesystem

The final step is to mount the new filesystem in the filesystem tree. There must first be a directory at the mount point. Use the **mount -a** command to check the `/etc/fstab` and mount the new filesystem. This also checks for errors that might prevent your system from booting properly.



# Device Recognition

- Master Boot Record ( MBR ) contains:
  - Executable code to load operating system
  - Space for partition table information, including:
    - Partition id or type
    - Starting cylinder for partition
    - Number of cylinders for partition



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

When a system boots, a search for code that can start an operating system is made from a list of devices (e.g. CD-ROM, floppy, hard drive) as defined in the BIOS. The first executable code found is used. Most often, a system boots from a hard drive attached to the system mainboard, and from the code found at the first sector, of the first cylinder--the *Master Boot Record*, or MBR--of that drive. This executable code is called a *boot loader*.

There are many boot loaders: their function is the same, but their interfaces and capabilities differ. The original MS-DOS boot loader has limited capabilities and is only capable of booting Microsoft DOS and older versions of the Windows operating system. Other boot loaders such as the one used by Microsoft WindowsNT/2000, or GNU GRUB (GRand Unified Boot loader) are more flexible, and can accommodate systems configured to run more than one operating system. Note that boot loaders do not load more than one OS at a time, even on *multi-boot* systems.

## Primary Partitions

In addition to the boot loader described above, the MBR contains a structure describing the hard drive partitions. IDE drives on Linux use legacy structures to describe four primary partitions, with provision for an extended partition and its logical partitions. Each partition, described by its size -- in sectors, blocks, or cylinders, and its offset from the zeroth cylinder, has a type which is also stored in this MBR structure. Linux-specific partitions would normally be one of the following types:

- 5 (or f) -- Extended
- 82 -- Linux swap
- 83 -- Linux
- 8e -- Linux LVM
- fd -- Linux RAID auto



# Disk Partitioning

- An extended partition points to additional partition descriptors
- Total maximum number of partitions supported by the kernel:
  - 63 for IDE drives
  - 15 for SCSI drives
- Why partition drives?
  - containment, performance, quotas, recovery



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Extended Partitions and Logical Partitions

If one of the primary partitions is marked as Extended (type '0x5') or Win95 Extended (type '0xf'), then the first sector of the partition described by the entry will contain another block of partition descriptors. These descriptors define partitions known as logical partitions. Use of logical partitions is a work-around for limitations in the legacy, Microsoft DOS-based partition table structure. Logical partitions permit the definition of more than four partitions per drive. While the PC partition specification does not impose a limit on the number of logical partitions, the kernel does.

## Partition Limits

The Linux kernel is designed with specific device numbers, the numeric name of the device driver for a given device. This allocation of device and number supports a maximum of 63 total partitions on each IDE disk, with one partition assigned one device number. On SCSI disks, the maximum number of partitions supported is 15, again due strictly to device number allocation. For more information about the kernel, install the `kernel-doc` RPM and reference `/usr/share/doc/kernel-doc-*/` for, among others, `devices.txt`.

## Why partition?

Unix best practices suggest that we should partition our disks for many reasons. By creating a separate filesystem we can contain applications and users to that filesystem. If it fills up because of security breach or user demand, the rest of the operating system is more insulated from the issue. Separate partitions improve performance by keeping data together which reduces disk head seek. If you would like to use quotas, they are enabled at the filesystem level. Also partitioning eases backup and recovery. If your application and its data are on separate filesystems, the operating system can be upgraded or reinstalled without having to restore the data from elsewhere.

# Managing Partitions

- Create partitions using:
  - **fdisk**
  - **sfdisk**
  - GNU **parted** - advanced partition manipulation (create, copy, resize, etc.)
- **partprobe** - reinitializes the kernel's in-memory version of the partition table



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## fdisk

According to its own documentation, **fdisk** is a buggy program that does fuzzy things-- usually it happens to produce reasonable results. The man page is too critical-- **fdisk** is the most commonly used partitioning program. It has the advantage of including some support for BSD disk labels and other non-DOS partition tables.

## sfdisk

The **sfdisk** user interface is somewhat cryptic, but it is more accurate than **fdisk** as well as more flexible. Moreover, it can be used non-interactively (that is, in a script).

## GNU parted

After installation you may need a program for creating, removing, resizing, and copying partitions containing filesystems. **parted** manages these tasks for a variety of filesystem types.

## partprobe

At system bootup, the kernel makes its own in-memory copy of the partition tables from the disks. Most tools like **fdisk** edit the on-disk copy of the partition tables. To update the in-memory copies, run **partprobe**.



# Making Filesystems

- **mkfs**
- **mkfs.ext2**, **mkfs.ext3**, **mkfs.msdos**
- Specific filesystem utilities can be called directly
  - **mke2fs** [*options*] *device*



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The command for creating a filesystem is **mkfs**. This command is a front end or wrapper for various filesystem creation programs. **mkfs -t *fstype*** looks for programs that follow the naming convention **mkfs.*fstype***, and then runs the program to create the desired filesystem. If you run **mkfs** without the **-t** option, it assumes the Linux default ext2 (Second Extended) filesystem. The **mkfs.*fstype*** form of the program can also be called directly. The ext2 filesystem has an additional interface called by:

## **mke2fs** [*options*] *device*

Some useful options include:

**-b** Specify the size of datablocks in bytes. Without this option, the datablock size is determined by the size of the partition. Filesystems that will contain many small files, for example, should use smaller block sizes because each file uses an entire block, even if its data is less than the size of one block. One file occupies, at least, one datablock. Compare the variable size of a datablock with the invariable size of the physical hard drive sector (usually 512 bytes).

**-c** Check the device for bad blocks (sectors) before creating the filesystem. This can take several hours if the partition is very large.

**-i** Specify the bytes/inode ratio. **mke2fs** creates an inode table based on the total size in bytes of a potential filesystem. Large datasets managed by a filesystem benefit from a higher ratio as there are fewer inodes, freeing more data space. Only one inode is allocated per file, while one inode may reference one or more datablocks. Inodes are 128 bytes in size, and unused inodes can occupy a lot of available space.

**-N** Override the default calculation of how many inodes are reserved for the filesystem. By default, this is based on the number of blocks and the bytes/inode ratio. The **-N** option allows you to specify the number of inodes directly. *Note:* This may be a more effective method of utilizing the partition or dataspace bytes/inode ratio for extremely large datasets, as the largest value passed to the **-i** option above is 8192.

**-m** Specify the percentage of reserved blocks for the superuser. This value defaults to 5%. If the filesystem being created is for a specific application, changing this percentage to zero allows the application full use of the filesystem. However, the filesystem also uses the reserved blocks to avoid *fragmentation*. Fragmentation happens when there are not enough blocks available to save a file completely within one *block group*. A highly fragmented filesystem performs poorly. The only way to remove fragmentation is to recreate the filesystem and restore the data from backup.

**-L** Set the volume label for the filesystem. This option's usefulness becomes apparent later when we discuss how filesystems are connected together on a Red Hat Enterprise Linux system.

**-j** Create an ext3 journal inode and filesystem. This is default for ext3.



# Filesystem Labels

- Alternate way to refer to devices
- Device independent
  - `e2label special_dev_file [fslabel]`
  - `mount [options] LABEL=fslabel mount_point`
- **blkid** can be used to see labels and filesystem type of all devices.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A potential problem exists when using special device files to point to file systems in that if the device is somehow relocated on the system, then the special device file that points to it will change. This most commonly happens with SCSI devices. Filesystem labels provide an alternate way to reference file systems for mounting that is not dependent on the special device file.

Two mechanisms exist to support filesystem labels. A filesystem label can be written into the superblock of ext2/ext3 filesystems using the **e2label** command:

```
e2label /dev/hda7 dbdisk1
```

Would create a label of dbdisk1 on the filesystem on partition /dev/hda7. The command:

```
e2label /dev/hda7
```

will display the current filesystem label for that device. The filesystem can be mounted using the command:

```
mount LABEL=dbdisk1 /mnt/data
```

# tune2fs

- adjusts filesystem parameters
  - reserved blocks
  - default mount options
  - **fsck** frequency
- View current settings with **dumpe2fs**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**tune2fs** sets parameters for ext2 and ext3 filesystems. Possible settings include:

- Modify the percentage of reserved blocks. *Note:* values lower than 5% can lead to performance degradation due to fragmentation

```
tune2fs -m 10 /dev/sda1
```

- Set the default mount options:

```
tune2fs -o acl,user_xattr /dev/sda1
```

- Disable mandatory filesystem checks

```
tune2fs -i0 -c0 /dev/sda1
```

To view the current settings use the **dumpe2fs** command. This command also displays the layout of the filesystem.

more 5 fsck



# Mount Points and /etc/fstab

- Configuration of the filesystem hierarchy
- Used by **mount**, **fsck**, and other programs
- Maintains the hierarchy between system reboots
- May use filesystem volume labels in the device field
- The **mount -a** command can be used to mount all filesystems listed in the /etc/fstab



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

/etc/fstab is referenced each time the system boots to create the desired filesystem hierarchy. It consists of six fields per line for each filesystem to be connected to the tree as follows:

| # device        | mount_point | FS_type | options  | dump_freq | fsck_order |
|-----------------|-------------|---------|----------|-----------|------------|
| LABEL=/mnt/data | /mnt/data   | ext3    | defaults | 0         | 0          |

**device** -- The special device file name, or filesystem label of the device to mount

**mount\_point** -- the path used to access the filesystem

**FS\_type** -- The filesystem type

**options** -- A comma-separated list of options, discussed later as options to **mount**

**dump\_freq** -- Level 0 dump frequency: 1=daily, 2=every other day, etc.; 0=never dump

**fsck\_order** -- 0 = ignore, 1 = first (the root filesystem should have this value), 2-9 = second, third, etc.: filesystems that have the same number greater than 1 are checked in parallel. Network filesystems and CD-ROMs should be ignored.

Using /etc/fstab when creating new mount points: After the filesystem has been created on a device and the filesystem has been labeled create an entry in the fstab that shows where it should be mounted. Important: After the entry in the /etc/fstab has been created use the **mount -a** to mount the filesystem just created and check for errors. It's better to discover an error when the system is online then to discover it later. In many cases errors in the /etc/fstab will cause **rc.sysinit** to fail and boot to **sulogin** (also known as emergency mode).

During system initialization (see /etc/rc.d/rc.sysinit), /etc/fstab is used to create the filesystem hierarchy. Entries are parsed and used as arguments to mount if, among their options, noauto is not present. At best, /etc/fstab is a wish list. If a filesystem is not available after reaching the desired runlevel (discussed later) run mount without options to display what is available. Floppy and CD-ROM entries typically have noauto as an option.

Using /etc/fstab after system initialization, an example: If the partition /dev/hda5 contained a filesystem labeled /mnt/data and the directory /mnt/data exists (and is already available!), then the following mount commands will connect this filesystem to the filesystem tree, using /etc/fstab to provide specifics:

```
mount /dev/hda5
```

```
mount -L /mnt/data
```



**# mount LABEL=/mnt/data**

**# mount /mnt/data**

# Mounting Filesystems with mount

- **mount [options] device mount\_point**
  - -t *fstype* (normally not needed)
  - -o *options*
  - Default options: rw, suid, dev, exec, acl, and async



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

As mentioned earlier, in order to access files of an individual filesystem, it must first be connected to the filesystem tree. This is done with the '**mount**' command.

```
mount [-t fstype] [options] device mount_point
```

If the mount command is invoked without any arguments, it reads the file `/etc/mtab`, maintained by the system, to display the currently mounted filesystems

The mount command takes several arguments to indicate the type of filesystem to be mounted and any special options that should be set (such as read-only access.) A useful reference to available options is the man page for **mount**. The filesystem type must precede the options because a given option may be specific to the filesystem type. One or both of the arguments for device and mount point must be used. If only one is used, then an entry in `/etc/fstab` must exist to supply the remaining information.

## Default Options for ext3

|       |                                     |
|-------|-------------------------------------|
| rw    | read and write access               |
| suid  | suid or sgid file modes honored     |
| dev   | device files permitted              |
| exec  | permit execution of binaries        |
| async | File changes managed asynchronously |
| acl   | Posix ACLs are honored              |

## Other options

|                         |                                                                                                                                 |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| uid=henry,<br>gid=henry | All files of the mounted filesystem are owned by, in this example, henry.                                                       |
| loop                    | mount the filesystem using a loopback device. Helpful when mounting a filesystem when it is a file of another filesystem.       |
| owner                   | similar to the user option, but in this case the mount request and the device, or special file, must be owned by the same EUID. |



# Unmounting Filesystems

- **umount** [*options*] *device* | *mount\_point*
- You cannot unmount a filesystem that is in use
  - Use **fuser** to check and/or kill processes
- Use the **remount** option to change a mounted filesystem's options atomically
  - **mount -o remount,ro /data**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

There are a few reasons for unmounting or disconnecting a filesystem from the root hierarchy. When you bring the system offline, reboot it, or perform filesystem maintenance, or when you use removable media, disconnect filesystems with **umount**. This command references */etc/mtab*. You can run **umount -a** (only as superuser) to disconnect all filesystems, or run:

```
umount [options] device | mount_point
```

For reasons of operating system stability and protection, you cannot unmount a filesystem that has open files, file handles, or a process's CWD, or is otherwise in use. On a removable media device, the device driver attempts to lock the device. If successful, the lock is removed when the device is quiescent, when unused. This lock, or an ignored **umount** command, can be frustrating. Should you experience this, **fuser** is helpful.

**fuser** displays information about the processes using a filesystem. After determining what is acting on the filesystem, **fuser** also provides a convenient way to send signals to those processes. It can prompt you interactively for each process before sending a signal, or kill all processes acting on the filesystem, including a user's CWD entry. To display what (or who) is acting on filesystem, run:

```
fuser -v mount_point
```

To kill all actions on a filesystem, run:

```
fuser -km mount_point
```

Sometimes you need to change the options of a mounted filesystem atomically (that is, without other operations occurring during the change). For example, if the root filesystem is mounted read-only (as is common during recovery operations), and you want to edit a configuration file on the root filesystem, you must mount the filesystem read-write before you can edit the file. If the root filesystem is on */dev/hda5*, the following **mount** command reinitializes the root filesystem's mount in a single operation (atomically), using the new options:

```
mount -o remount,rw /dev/hda5 /
```



# mount By Example

- Sample filesystem requirements met using options:
  - Disabling execute access
  - Mounting a filesystem image
  - Mounting a PC-compatible filesystem
  - Disabling access time updates
  - Setting up a **mount** alias



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Listed below are a few **mount** commands and their explanations.

```
mount -t ext3 -o noexec /dev/hda7 /home
```

For security, users' home directories should be connected denying permission to execute files managed there.

```
mount -t iso9660 -o loop /iso/documents.iso /mnt/cdimage
```

Mount the CD-ROM image file `/iso/documents.iso` read-only using the first available loopback device. This provides access to a filesystem (in this case, organized as a common CD-ROM file structure) that is itself a file of another filesystem.

```
mount -t vfat -o uid=515,gid=520 /dev/hdc2 /mnt/projX
```

Mount the vfat filesystem located on the `/dev/hdc2` partition so that each file is owned by a specific UID and GID. This is convenient for the end-user whose UID and GID are as listed, who wants to manage this filesystem data. Normally, the filesystem's data would be owned by root (the superuser), denying direct file manipulation to others.

```
mount -t ext3 -o noatime /dev/hda2 /data
```

Mount the filesystem using the **noatime** option to increase I/O performance or laptop battery up-time by reducing disk access.

```
mount --bind /something /anotherthing
```

This command is available starting with the 2.4 kernel. This mounts a directory already mounted on the filesystem on another mount point.

# Handling Swap Files and Partitions

- Swap space is a supplement to system RAM
- Basic setup involves:
  - Create swap partition or file
  - Write special signature using **mkswap**
  - Add appropriate entries to `/etc/fstab`
  - Activate swap space with **swapon -a**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Setting up a swap partition

Use a partitioning program to add a partition. Set the partition id type to 0x82. Create the signature needed on the partition using **mkswap**:

**mkswap /dev/hda6**

Add an entry for the swap to `/etc/fstab`. It will look similar to the following:

```
/dev/hda6 swap swap defaults 0 0
```

Activate the swap partition using **swapon -a** (which reads `/etc/fstab` and turns on all swap entries it lists).

Check the swap partition's status using **swapon -s**.

## Setting up a swap file

Use the following to create a file, where count X defines the file size in kilobyte blocks:

**dd if=/dev/zero of=/swapfile bs=1024 count=X**

Run **mkswap** to create the signature. The swapfile can also be activated manually with **swapon**, or added entry to `/etc/fstab`:

```
/swapfile swap swap defaults 0 0
```

*part probe*

*(create partition -x255)  
plug it after system in kernel  
(update) place /etc/fstab (2,1,1)  
note count must be empty, which  
was developed as /*



# Mounting NFS Filesystems

- Makes a remote NFS filesystem work as though it were a local filesystem
- `/etc/fstab` can be used to specify persistent network mounts
- NFS shares are mounted at boot time by `/etc/init.d/netfs`
- Exports can be mounted manually with the **mount** command.

```
mkdir /mnt/server1
mount -t nfs server1:/var/ftp/pub /mnt/server1
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

To associate a shared directory on the network with a mount point in your local filesystem, use **mount**. When you mount an exported directory from an NFS server, you can access it as if it was local to your machine. Shares can be mounted manually by **root**, or automatically at boot time.

`/etc/fstab` allows you to specify network directories to be mounted at boot. Here's a sample `fstab` entry that defines a shared filesystem `/var/ftp/pub` on `server1` to be mounted locally as `/mnt/server1`.

```
server1:/var/ftp/pub /mnt/server1 nfs defaults 0 0
```

`/etc/init.d/netfs` mounts any network filesystems that are configured to be mounted at boot time, such as the one defined above.

Some NFS-specific options that can be used with **mount** or in `/etc/fstab` include:

- `rsize=8192` and `wsiz=8192` - will speed up NFS throughput considerably
- `soft` - processes return with an error on a failed I/O attempt
- `hard` - will block a process that tries to access an unreachable share
- `intr` - allows NFS requests to be interrupted or killed if the server is unreachable
- `nolock` - disables file locking (**lockd**) and allows inter-operation with older NFS servers.



# Automounter

- System administrator specifies mount points controlled by automounter daemon process in `/etc/auto.master`
- The automounter monitors access to these directories and mounts the filesystem on demand
- Filesystems automatically unmounted after a specified interval of inactivity
- Enable the special map `-host` to “browse” all NFS exports on the network
- Supports wildcard directory names



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

By default, all mounted file systems are owned by the root account and can only be unmounted by the root account. This behavior can be overridden through the use of the `owner`, `user` or `users` options in `/etc/fstab`. This is normally done, for example, for removable media devices so that a non-privileged user can access the contents of a floppy disk or cdrom.

In addition to removable media devices, users may often need access to files located elsewhere on the network. This access can be obtained by mounting a network file share. Whether a network file share or some type of removable media device, one drawback is that casual users must learn the syntax of the mount and unmount commands. The automounter can be configured to monitor certain directories and automatically mount the appropriate devices when a reference is made to files in that directory. The automount daemon is provided by the `autofs` RPM.

To control filesystems with automount, modify the supplied `/etc/auto.master`.

The `/etc/rc.d/init.d/autofs` script parses this file and launches the automount daemon based on this configuration. Each line of `auto.master` lists a directory, present in the hierarchy, and a reference to yet another file that further defines specific mount options for those base mount points.

```
[root@stationX ~]# cat /etc/auto.master | grep "^/"
/misc /etc/auto.misc
/net -hosts
```

Referred from an `auto.master` entry, an `autofs` mount configuration file lists the filesystems to be mounted under the directory, including options required. Sample syntax can be found in the `autofs(5)` man page.

```
[root@stationX ~]# cat /etc/auto.misc | grep -v "#"
cd -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom
```

Once a mount point has been configured for automount, and the daemon started, merely requesting the filesystem (`ls`, `cd`, or other action) completes the mount. The daemon maintains the connection as long as the filesystem is in use plus an interval of time. The default interval is 60 seconds.

```
[root@stationX ~]# mount | grep cd
[root@stationX ~]# ls /misc
[root@stationX ~]# ls /misc/cd
autorun README-ja.html
```

RELEASE-NOTES-gu.html

EULA                      README-ko.html                      REALEASE-NOTES-hi.html  
... output truncated ...  
[root@stationX ~]# mount | grep cd  
/dev/sdc on /misc/cd type iso9660 (ro,nosuid,nodev)

Autofs mount configuration files support a wildcard system that allows multiple mounts to be specified within a directory. This is commonly used to mount home directories. If the following entry were in `/etc/auto.home.guests`, any directory within `server1:/home/guests` when the corresponding directory within `/home/guests` on the local system is accessed:

```
* -fstype=nfs server1:/home/guests/&
```

A special `-hosts` map can be configured in `/etc/auto.master` to browse NFS shares provided by other systems. The `-hosts` map only mount shares when accessed (lazy mounting), greatly reducing the load big file servers.

Browsing (also called ghosting) can be enabled globally in `/etc/sysconfig/autofs` by setting `DEFAULT_BROWSE_MODE="yes"` or by putting `-g` after the entry in `/etc/auto.master`.

## Direct Maps

- Direct maps include absolute path names
- Does not obscure local directory structure
- Example:

```
/etc/auto.master:
/- /etc/auto.direct
```

```
/etc/auto.direct:
/foo server1:/export/foo
/usr/local/ server1:/usr/local
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Direct maps can be used to automatically mount file systems to any mount point in the filesystem. A direct map contains absolute path names and is referenced with `/-` in `/etc/auto.master`.

Only the right-most directory is put under automounter control. Thus the directory structure above the mount point (`/usr` in this example) is not obscured by **autofs**.



# gnome-mount

- automatically mounts removable devices
- integrated with the HAL (Hardware Abstraction Layer)
- replaces fstab-sync



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**gnome-mount** automatically mounts removable devices when they are detected via HAL (Hardware Abstraction Layer). This replaces fstab-sync which was used for Red Hat Enterprise Linux 4. Configuration - though rarely necessary - is handled by Gnome's Gconf system.

## End of Unit 6

- Questions and Answers
- Summary
  - What tools are available for partitioning?
  - What two ways can swap space be implemented?
  - Mount NFS

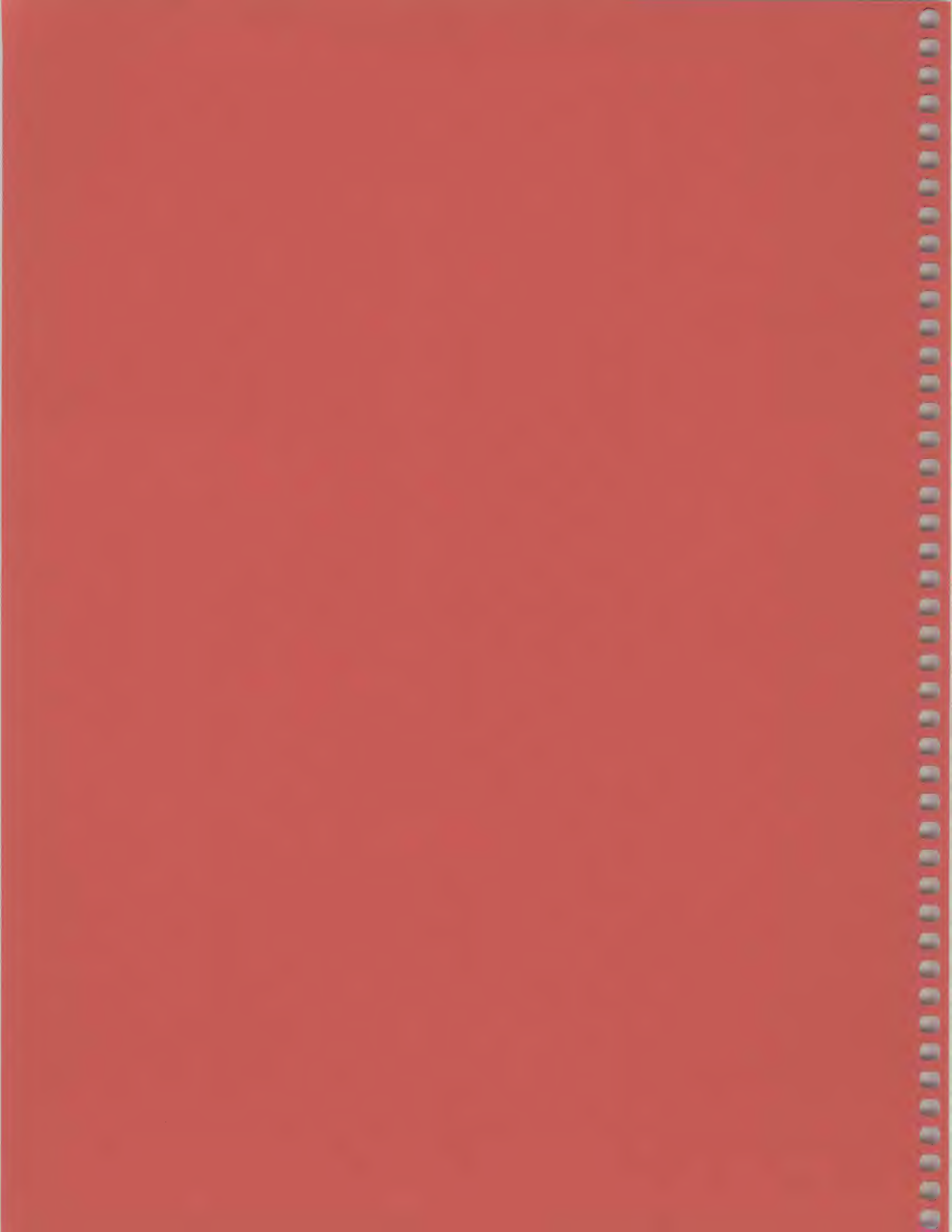


For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.









# Lab 6

## Adding New Filesystems to the Filesystem Tree

---

Goal: Develop skills and knowledge related to partitioning and filesystems

Estimated Duration: 120 minutes



## Sequence 1: Creating and Mounting Filesystems

### Instructions:

1. Use **fdisk -l** to locate information about your harddisk. Depending on hardware you will see either a `/dev/sda` or `/dev/hda` disk. Partition sizes are given in cylinders. Notice the total number of cylinders and the End cylinders of the extended partition and last logical partition. You should see that the extended partition spans until the end of the drive, while the last logical partition ends much earlier.
2. Now use **fdisk** to add a new logical partition that is 1GB to 2GB in size. (Make sure to write the changes to disk using the **w** command.). Run **fdisk -l** to verify that the partition was created.
3. Run **cat /proc/partitions**. Why does the new partition not show up?  

*no re boot yet*

---
4. Reboot to reread the revised partition table or use **partprobe** to refresh the kernel's view of the partition table.
5. Create an ext3 filesystem on the new partition. Use 2 KB sized blocks and one inode per every 4 KB of disk space. Assign the label `opt` to the filesystem, mount it on the filesystem tree at `/opt`. Make sure that the system mounts this filesystem after reboot.
6. **CLEANUP:**
  - a. Unmount `/opt` and comment out the line in `/etc/fstab` that references `/opt`.
  - b. Use **fdisk** to delete the partition you just created. *Hint:* The last partition created will be the partition with the highest number.

## Sequence 2: Mounting an NFS Filesystem

### Instructions:

1. ✓ Ensure that iptables firewalling is disabled.
2. ✓ Mount the export `server1:/var/ftp/pub` persistently at `/mnt/server1`.

hda3

Start 1161

end 1225

## Sequence 3: Automounting data with autofs

### Instructions:

1. ✓ Ensure that iptables firewalling is disabled.
2. ✓ Configure automounter to mount `server1:/var/ftp/pub` to `/misc/server1`.
3. Use the automounter to mount the home directories for your NIS users from `server1.example.com`. You can use **getent passwd** to see what home directories are assigned. `server1.example.com` exports `/home/guests` to your system.



## Sequence 1 Solutions

2. Now use **fdisk** to add a new logical partition that is 1GB to 2GB in size. (Make sure to write the changes to disk using the **w** command.). Run **fdisk -l** to verify that the partition was created.

fdisk requires a single disk device as parameter. If your workstation is equipped with a SATA drive this is `/dev/sda`. Otherwise use `/dev/hda`.

```
fdisk /dev/hda
```

The following commands are all within **fdisk**:

- a. Within fdisk, type **p** to print the current partition table. Verify that the drive already has an extended partition. If there is none, create one that spans the rest of the disk now:
    - a. Press **n** for a new partition
    - b. **e** to create an extended partition
    - c. **4** as the partition number
    - d. First cylinder: First unused cylinder is default, so type *Enter* here
    - e. Last cylinder: Last cylinder of disk is default, press *Enter*
  - b. Create a new logical partition.
    - **n** for a new partition
    - **l** to create an logical partition (skipped if no primary partitions left)
    - First cylinder: *Enter*
    - Last cylinder: **+1024M**
  - c. Write the new partition table with **w**. This will also terminate the fdisk command
3. Run **cat /proc/partitions**. Why does the new partition not show up?

The kernel has not reread its partition table

4. Reboot to reread the revised partition table or use **partprobe** to refresh the kernel's view of the partition table.
5. Create an ext3 filesystem on the new partition. Use 2 KB sized blocks and one inode per every 4 KB of disk space. Assign the label `/data` to the filesystem. Make sure that the system mounts this filesystem after reboot.

The example solution uses the **mkfs.ext3** command. **mke2fs** with the **-j** option could have also been used. The **-L** option sets the filesystem label, we could have just as easily used a

separate call to **e2label** after we created the filesystem. As is the case with just about all our examples, there is more than one way to do it.

a. `# mkfs.ext3 -L opt -b 2048 -i 4096 device`

b. `# mkdir /data`

c. Edit `/etc/fstab`:

```
LABEL=opt /opt ext3 defaults 1 2
```

d. Use the **mount -a** command to mount the new file system on `/opt`. Were there any errors? If so check the syntax of the `/etc/fstab` and try again. Use the **mount** command, this time with no options, to confirm that `/opt` mounted properly. Copy `/etc/passwd` into `/opt` and verify that the copy was successful.

e. Reboot the system to verify that the filesystem is automatically mounted after boot.

## 6. *CLEANUP:*

a. Unmount `/opt` and comment out the line in `/etc/fstab` that references `/opt`.

```
umount /opt
```

b. Edit `/etc/fstab`:

```
#LABEL=opt /opt ext3 defaults 1 2
```

c. Use `fdisk` to delete the partition you just created. *Hint:* The last partition created will be the partition with the highest number.

Run `fdisk` on the disk device from earlier in the unit.

```
fdisk /dev/hda
```

(Following commands are all within `fdisk`.)

d. Delete the last partition on the list.

- `d` for a delete partition
- Choose the highest number listed. For example, if it says:

```
Partition Number (1-9):
```

```
Choose 9
```

- Write the new partition table with `w`. This will also terminate the `fdisk` command

- e. Run **partprobe** to refresh the kernel's view of the partition table.



## Sequence 2 Solutions

1. Ensure that iptables firewalling is disabled.

```
service iptables stop
chkconfig iptables off
```

2. Mount the export `server1:/var/ftp/pub` persistently at `/mnt/server1`.

- a. `# mkdir /mnt/server1`

- b. Edit `/etc/fstab`:

```
server1:/var/ftp/pub /mnt/server1 nfs defaults 0 0
```

- c. Use the **mount -a** command to mount the export on `/mnt/server1` and check for errors in the `/etc/fstab`

- d. Reboot your system to test.

```
init 6
```

## Sequence 3 Solutions

1. Ensure that iptables firewalling is disabled.

```
service iptables stop
chkconfig iptables off
```

2. Configure automounter to mount `server1:/var/ftp/pub` to `/misc/server1`.

- a. Edit the `/etc/auto.master` file. Uncomment the first line for `/misc`.
- b. Add a line to the `/etc/auto.misc` file that will mount the `/var/ftp/pub` export from `server1.example.com` to your own `/misc/server1` target:

```
server1 -ro,intr,hard server1:/var/ftp/pub
```

- c. Restart the autofs service:

```
service autofs restart
```

- d. Try to access the `/misc/server1` directory.

3. Use the automounter to mount the home directories for your NIS users from `server1.example.com`. You can use **getent passwd** to see what home directories are assigned. `server1.example.com` exports `/home/guests` to your system.

- a. Begin by editing `/etc/auto.master` and add the following line:

```
/home/guests /etc/auto.guests --timeout=60
```

This line specifies that `/etc/auto.guests` defines mount points in `/home/guests` managed by the automounter. When not in use for more than 60 seconds, filesystems mounted on those mount points are automatically unmounted.

- b. Create and edit `/etc/auto.guests` so it contains the line

```
* -rw,soft,intr 192.168.0.254:/home/guests/&
```

This line specifies that access to any immediate subdirectory of `/home/guests` should make autofs mount a NFS export from `192.168.0.254` where the `&` is the same as the name of the local subdirectory. (So the automounter would mount `192.168.0.254:/home/guests/guest2001` on `/home/guests/guest2001`). The middle column specifies the mount options that will be used; read-write, timeout eventually if the NFS server is not available, and timeout immediately if an interrupt is sent.

- c. Configure **autofs** to start in run levels 2, 3, 4, and 5, then restart it manually:

```
[root@stationX]# chkconfig autofs on ; service autofs restart
```

- d. Now try logging in again and see whether the home directory gets mounted automatically. It should. Try logging into to your neighbors system once it is also configured. You should be able to access your home environment from any system in the `notexample` domain.







## Unit 7

# Advanced Filesystem Management



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[training@redhat.com](mailto:training@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



# Objectives

Upon completion of this unit, you should be able to:

- Setup filesystem quotas
- Setup and manage software Raid devices
- Configure Logical Volumes
- setup LVM Snapshots
- perform backups



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Configuring the Quota System

- Overview
  - Implemented within the kernel
  - Enabled on a per-filesystem basis
  - Individual policies for groups or users
    - Limit by the number of blocks or inodes
    - Implement both soft and hard limits
- Initialization
  - Partition mount options: `usrquota`, `grpquota`
  - Initialize database: **quotacheck**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Linux quota system allows an administrator to establish limits on the amount of disk resources users can consume. Because resource accounting must occur with every file creation, quotas are implemented within the kernel. Various required quota administration utilities are found within the quota RPM.

## Quota initialization

The following examples show how to implement user quotas on the `/home` partition. Group quotas are implemented in a nearly identical manner.

- Define partition options:

For a partition to implement quotas, it must be mounted with the `usrquota` or `grpquota` options. These options can be added to the appropriate entries in `/etc/fstab`. After you edit the file, the options can be made to immediately take effect by remounting the filesystem.

Edit `/etc/fstab`, adding the `usrquota` option to the `/home` partition:

```
mount -o remount /home
```

- Create/Update the database:

The disk usage database is stored within a partition's top-level directory in specially named binary files, `aquota.user` and `aquota.group`. Use **quotacheck -cug** to create a new user and group quota file. During initialization, or any time the database file is out of sync with the actual state of a partition (for example, if quotas were turned off for a period of time, or if a system was brought down ungracefully, without unmounting partitions), the database can be brought up to date by running the **quotacheck** command:

```
quotacheck -c /home
```



# Setting Quotas for Users

- Implementation
  - Start or stop quotas: **quotaon**, **quotaoff**
  - Edit quotas directly: **edquota** *username*
  - From a shell:

```
setquota username 4096 5120 40 50 /foo
```

- Define prototypical users:

```
edquota -p user1 user2
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

- Starting and stopping quotas:

Quotas are turned on and off with the **quotaon** and **quotaoff** commands. These commands either take partitions as their arguments, or are invoked with the **-a** command line switch, in which case quotas are turned on for all appropriate partitions defined in `/etc/fstab`. These commands rarely need to be run in practice, because they are included within the default RHEL initialization script `/etc/rc.d/rc.sysinit`.

```
quotaon /home
```

```
quotaon -a /home
```

- Editing user policies:

User policies are implemented with the **edquota** command. This command invokes an editor and loads a template, which can then be edited to establish the appropriate values. These values are committed to the database when you exit the editor. To ease the propagation of quotas, user policies can be prototyped from established policies for another user. Often, when first establishing quotas, it is helpful to first define a prototypical user, and then edit the user's properties. (If you do not first prototype the user, no template is provided by the **edquota** command.) Grace periods are established by using the **-t** command line switch with **edquota**.

```
edquota user1
```

(implement a policy for *user1*)

```
edquota -p user1 user2
```

(mimic the policy for *user2* from the policy for *user1*)

```
edquota -t
```

(establish a grace period)



# Reporting Quota Status

- Reporting
  - User inspection: **quota**
  - Quota overviews: **repquota**
  - Miscellaneous utilities: **warnquota**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

- Generating quota reports:

Users can inspect their disk usage and quotas by issuing the **quota** command. An administrator can generate a report of disk usages by all users with the **repquota** command. Users over their quotas can be warned with a **warnquota** cron job.

# What is Software RAID?

- Multiple disks grouped together into "arrays" to provide better performance, redundancy or both.
- **mdadm** - provides the administration interface to software RAID.
- Many "RAID Levels" supported, including RAID 0, 1 and 5.
- Spare disks add extra redundancy
- RAID devices are named, `/dev/md0`, `/dev/md1`, `/dev/md2`, `/dev/md3` and so on.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Either partitions or whole disks can be used to create software RAID devices. The term "disk" is used here to denote either a whole disk or a partition one or more disks. Though it's possible to create an array composed entirely of partitions on a single device, as is demonstrated in this class, there's little benefit in doing so outside of the lab environment.

The most commonly used RAID types are:

- **RAID 0 or Striping:** Two or more disks used to create a single large high performance volume. Performance is better if drives of equal size are used. No redundancy, so chance of failure is very high. Array size equals the sum of all disks in array.
- **RAID 1 or Mirroring:** Two disks containing the the same data updated simultaneously. Redundancy offers good protection against disk failure. Can slow write performance but tends to improve read performance. Only RAID type that you can place the `/boot` partition on. Hot spare disks can be used to improve fault-tolerance. Array size equals the size of the smallest disk used.
- **RAID 5:** Three or more disks with zero or more hot spares. A good balance between performance and reliability. Redundancy is achieved by splitting parity between all disks, one disk can be lost without causing array failure. Both read and write speeds are usually improved, but in certain cases write performance is dramatically decreased. For this reason RAID 5 is often not a good choice to host databases.

Once an array is created its device, `/dev/md0` for example, is used the same way as `/dev/hda6` or `/dev/sda6` may have been used in an earlier example.



# Software RAID Configuration

- Create and define RAID devices using **mdadm**

```
mdadm -C /dev/md0 -a yes -l 1 -n 2 -x 1 /dev/sda1 /dev/sdb1 ✓
/dev/sdc1
```

- Format each RAID device with a filesystem

```
mke2fs -j /dev/md0
```

- Test the RAID devices
- **mdadm** allows you to check the status of your RAID devices

```
mdadm --detail /dev/md0
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

To install and configure software RAID after installation, the **mdadm** RPM is required. It should already be installed by default, since it is included in the minimal Core installer component.

1. Use **fdisk** or another tool to create disk partitions of type 0xfd, Linux RAID.
2. If necessary, run **partprobe** to have the kernel reload the partition table.
3. Initialize and activate your RAID array:

```
mdadm -C /dev/md0 --chunk=64 --level=5 --raid-devices=3 /dev/sd{b,c,d}1

--level or -l sets the RAID level. --raid-devices or -n sets the number of RAID disks.
--spare-device or -x optionally sets the number of hot spares. -a yes creates the device file if they do
not exist.
```

4. Create a filesystem on the new software RAID device. With **mke2fs**, there is a special **-R stride=n** option that can improve performance. The stride is the software RAID device's chunk-size in filesystem blocks. For example, with an ext3 filesystem that will have a 4 KB block size on a RAID device with a chunk-size of 64 KB, the stride should be set to 16:

```
mke2fs -j -b 4096 -R stride=16 /dev/md0
```

5. Add the software RAID device and its mount point to **/etc/fstab** as appropriate.



# Software RAID Testing and Recovery

- Simulating disk failures

```
mdadm /dev/md0 -f /dev/sda1
```

- Recovering from a software RAID disk failure

- replace the failed hard drive and power on
- reconstruct partitions on the replacement drive
- `mdadm /dev/md0 -a /dev/sda1`

- `mdadm`, `/proc/mdstat`, and syslog messages



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `mdadm -f` command can be used to simulate a drive failure. This is useful when testing RAID 1 or RAID 5 arrays, but will destroy a RAID 0 array. Spare disks can be set aside in a RAID 1 or RAID 5 array, that can automatically start rebuilding as a replacement disk ( `-x` option ). The array can be used while it is rebuilding, although performance will be degraded.

Recovery is fairly simple. Power off the system, replace the failed disk, power on the system, and reconstruct an appropriate partition table on the disk. The new disk can be inserted into the array with the `mdadm -a` command. You may have to remove the failed disk from the array with `mdadm -r`.

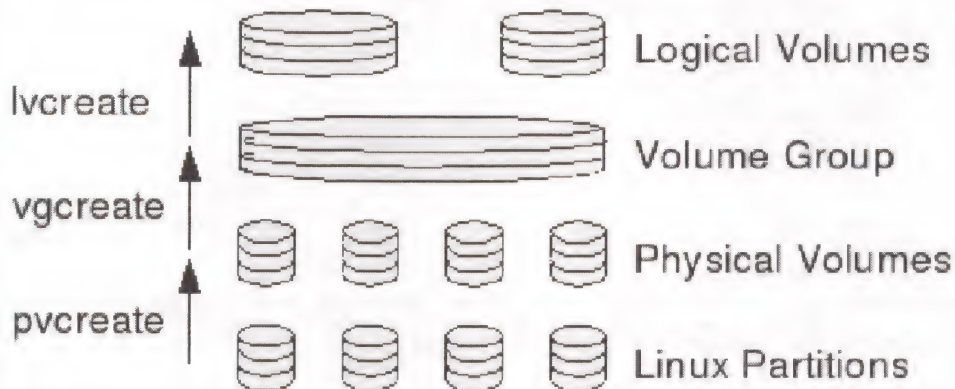
Information about disk failures is logged through syslog to `/var/log/messages` by default. Information on the current state of software RAID devices is also available in `/proc/mdstat`. The output below shows a RAID 5 device, `/dev/md0`, made up of `/dev/sdb1`, `/dev/sdc1`, and `/dev/sdd1`; `/dev/sdd1` has failed.

```
[root@stationX ~]# cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 sdd1[3] (F) sdc1[1] sdb1[0]
272896 blocks level 5, 64k chunk, algorithm 2 [3/2] [UU_]
unused devices: <none>
```

Information on estimated time to reconstruction will appear in this file if the array is rebuilding. You can also use the `mdadm --detail` command to view the status of the RAID array.

# What is Logical Volume Manager (LVM)?

- A layer of abstraction that allows easy manipulation of volumes. Including resizing of filesystems
- Allows reorganization of filesystems across multiple physical devices
  - Devices are designated as Physical Volumes
  - One or more Physical Volumes are used to create a Volume Group
  - Physical Volumes are defined with Physical Extents of a fixed size
  - Logical Volumes are created on Physical Volumes and are composed of Physical Extents
  - Filesystems may be created on Logical Volumes



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

LVM creates a higher-level layer of abstraction than traditional linux disks and partitions. This allows for great flexibility in allocating storage. Logical volumes can be resized and moved between physical devices easily. Physical devices can be added and removed with relative ease. LVM managed volumes can also have sensible names like "database" or "home" rather than the somewhat cryptic "sda" or "hda" device names.



# Creating Logical Volumes

- Create physical volumes

```
pvccreate /dev/hda3
```

- Assign physical volumes to volume groups

```
vgcreate vg0 /dev/hda3
```

- Create logical volumes from volume groups

```
lvcreate -L 256M -n data vg0
mke2fs -j /dev/vg0/data
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Any disk partitions to be used as physical volumes need to have their partition types set to 0x8e, Linux LVM. Then the partitions need to be added as physical volumes with `pvccreate device-name`. Software RAID devices may also be set up as physical volumes.

New volume groups need to be created and one or more physical volumes assigned to them with the `vgcreate` command. The new volume group can have almost any name. At this time the size of an extent is determined; by default it is 4 MB. This affects the minimum size of changes which can be made to a logical volume in the volume group, and the maximum size of logical and physical volumes in the volume group. A logical volume can contain at most 65534 extents, so the default extent size limits the volume to about 256 GB; a size of 1 TB would require extents of at least 16 MB:

```
vgcreate -s 16M vg0 /dev/hda3 /dev/sda2
```

Logical volumes are created from these volume groups, and may have arbitrary names. The size of the new volume may be requested in either extents or in KB, MB, GB, or TB (rounding up to whole extents):

```
lvcreate -L 512M -n data vg0
```

(asks for /dev/vg0/data of 512 MB)

```
lvcreate -l 32 -n test vg0
```

(asks for /dev/vg0/test of 32 extents)

Logical volumes may also be striped like a RAID 0 device between multiple physical volumes. A striped logical volume may be extended later, but only with extents from the original physical volumes. Because LVM can not tell if two particular physical volumes in the volume group are on the same drive, it is best not to do this if a volume group will contain striped logical volumes. The following command creates a logical volume, `/dev/vg1/striped`, striped between two physical volumes in the `vg1` volume group, using the default stripe size.

```
lvcreate -i 2 -L 1G -n striped vg1
```



# Resizing Logical Volumes

- Growing Volumes
  - **lvextend** can grow logical volumes
  - **resize2fs** can grow EXT3 filesystems online
  - **vgextend** adds new physical volumes to an existing volume group.
- Shrinking volumes
  - Filesystem must be reduced first
  - Requires a filesystem check and cannot be performed online
  - **lvreduce** can then reduce the volume.
  - Volume Groups can be reduced with:

```
pvmove /dev/hda3
vgreduce vg0 /dev/hda3
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Logical volumes can be resized dynamically while preserving the data on the volume, if the volume's filesystem supports resizing. The **lvextend** command allows resizing of an e or ext3-based logical volume. **resize2fs** can be used to grow mounted ext2 and ext3 filesystems. **lvextend** must be called first to grow the logical volume.

The following commands grow the mounted `/dev/vg0/data` filesystem:

```
lvextend -L +500M /dev/vg0/data
resize2fs /dev/vg0/data
```

For other filesystems, the **lvextend** utility can be used to add unallocated extents in the volume group to a logical volume. Then native utilities for the filesystem can be used to expand it to fill the volume. To reduce a filesystem, first use the native utilities to shrink the filesystem, then run **lvreduce** to shrink the logical volume.

Additional physical volumes can be added to a volume group to provide more unallocated extents to assign to logical volumes. The physical volumes need to be setup with **pvcreate**, then added to the volume group with the **vgextend** command.

Physical volumes can also be removed from a volume group. This is useful for removing an old disk from the volume group. The **pvmove** command can redistribute extents from the physical volume being decommissioned to the other physical volumes in the volume group. In its simplest mode, **pvmove** takes the name of the physical volume to be removed as its argument; for example:

```
pvmove /dev/hda3
```

Once there are no extents in use on the old physical volume, it can be removed from the volume group with the **vgreduce** command:

```
vgreduce vg0 /dev/hda3
```

Some commands are available to help you gather information about the state of your physical volumes, volume groups, and logical volumes. Three useful commands are **pvdisplay**, **vgdisplay**, and **lvdisplay**.

# Logical Volume Manager Snapshots

- *Snapshots* are special Logical Volumes that are an exact copy of an existing Logical Volume at the time the snapshot is created
- Snapshots are perfect for backups and other operations where a temporary copy of an existing dataset is needed
- Snapshots only consume space where they are *different* from the original Logical Volume
  - Snapshots are allocated space at creation but do not use it until changes are made to the original Logical Volume or the Snapshot
  - When data is changed on the original Logical Volume the older data is copied to the Snapshot
  - Snapshots contain only data that has changed on the original Logical Volume or the Snapshot since the Snapshot was created.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Common uses for snapshots include:

- Backup of live data: It's important that when taking a backup that the dataset in question not change while the backup is in progress. This can lead to inconsistencies and data loss.

When a snapshot is taken a static copy of the data from a live dataset is then available. This can be used to make a backup.

In the case of database software a common practice is to *quiesce* the database so that the data is in a consistent state before a snapshot is taken.

- Application Testing: During software development it may be necessary to test an application on the exact same data repeatedly. Snapshots allow for quick creation of multiple copies of a logical volume that don't consume as much space as they would otherwise.
- Hosting of Virtualized Machines: Once a single template machine is created snapshots can be used for individual Virtual Machines. This will save space in that many files will be identical on multiple machines.



# Using LVM Snapshots

- Create Snapshot of existing Logical Volume

```
lvcreate -l 64 -s -n databackup /dev/vg0/data
```

- Mount Snapshot

```
mkdir -p /mnt/databackup
mount -o ro /dev/vg0/databackup /mnt/databackup
```

- Remove Snapshot

```
umount /mnt/databackup
lvremove /dev/vg0/databackup
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Snapshot can be created in read-only mode:

```
lvcreate -l 64 -p r -s databackup /dev/vg0/data
```

Snapshot size can be assigned in human readable format rather than by Physical Extents.

```
lvcreate -L 512M -s databackup /dev/vg0/data
```

Use the **lvdisplay** to determine how much of the allocated space the Snapshot is using. Look for the Allocated to snapshot value to determine what percentage of the space allocated to the Snapshot is being used.

```
[root@stationX ~]# lvdisplay /dev/vg0/databackup
--- Logical volume ---
LV Name /dev/vg0/databackup
VG Name vg0
LV UUID ogeesl-ool1-kMZu-0o5b-DxmO-ge8T-GyLcue
LV Write Access read/write
LV snapshot status active destination for /dev/vg0/data
LV Status available
open 0
LV Size 512.00 MB
Current LE 64
COW-table size 128.00 MB
COW-table LE 4
Allocated to snapshot 20.00%
Snapshot chunk size 8.00 KB
Segments 1
Allocation inherit
Read ahead sectors 0
Block device 253:13
```

Snapshots can be expanded as other Logical Volumes can be.

```
lvextend -L +64M /dev/vg0/databackup
```

## Archiving tools: tar

- **tar** can backup to a file or tape device
- supports GZIP and BZIP2 compression
- can preserve file permissions, ownership and timestamps
- supports extended attributes
- uses **rmt** to write to a remote tape device



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The **tar** command requires one option to indicate the action to be taken. This could be **-c** to create a tar file, **-t** to list the table of contents of a tar file, or **-x** to extract files from the tar file. Typically, when creating a tar file or extracting contents from a tar file, you will want to see a list of files archived or extracted: use the **-v** option to see this. Finally, to specify the tar file or tape device, use the **-f** option.

It is typical, but not required, to compress a tar archive. The **tar** command can do this for you, using either the **gzip** compression tool with the **-z** option or the **bzip2** compression tool using the **-j** option.

When extracting files from a tar archive consider the **-p** option. This will preserve owner, permissions and timestamps when possible.

Current versions of tar support extended attributes. This requires the archive to be created with the **--xattrs** option. You can either backup SELINUX or ACLs by specifying **--selinux** or **--acls**. When extracting the archive use **--no-selinux**, **--no-acls**, or **--no-xattrs**



# Archiving Tools: dump/restore

- Back up and restore ext2/3 filesystems
  - Does not work with other filesystems
  - dump should only be used on unmounted filesystems or filesystems that are read-only.
- Can do full or incremental backups
- Examples:

```
dump -0u -f /dev/nst0 /dev/hda2
restore -rf /dev/nst0
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Using dump

The dump command can be directed to read `/etc/fstab` and do backups based on information it keeps as to which filesystems need to be backed up.

For example,

```
dump -0u -f /dev/nst1 /home
```

will do a full backup of the `/home` filesystem onto the tape device `nst1`. The `-u` option will update the file `/etc/dumpdates`, which will record dump information for future use by dump. After a level 0 backup, dump will perform an incremental backup every day on active filesystems listed in `/etc/fstab`.

The command:

```
dump -4u -f /dev/nst1 /home
```

will perform an incremental update of all files that have changed since the last backup of level 4 or lower, as recorded in the `/etc/dumpdates` file.

```
dump -0uf joe@svr:/dev/nst0 /home
```

performs a remote backup using **rmt**. Use `user@host:path` format to specify the remote user, host, and device. SSH can be used as a transport layer when `$RSH` is set to `ssh`.

## Using restore

To restore data backed up with dump, make a clean filesystem (using `mkfs`), mount the filesystem, and `cd` to the directory where the filesystem is mounted. Then run the restore command:

```
restore -rf /dev/nst1
```

## Archiving Tools: rsync:

- Efficiently copies files to or from remote systems
- Uses secure **ssh** connections for transport
  - **rsync \*.conf barney:/home/joe/configs/**
- Faster than **scp** - copies differences in like files



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**rsync** is a program that works in much the same way that the older **rcp** does, but has many more options and uses the **rsync** remote-update protocol to greatly increase the speed of file transfers when the destination file already exists.

The **rsync** remote-update protocol allows **rsync** to transfer just the differences between two sets of files using an efficient checksum-search algorithm. This utility is useful for tasks like updating web content, because it will only transfer the changed files.

### Useful options to **rsync**

|                   |                                                                                     |
|-------------------|-------------------------------------------------------------------------------------|
| <b>-e command</b> | specifies an external, rsh-compatible program to connect with (usually <b>ssh</b> ) |
| <b>-a</b>         | recurses subdirectories, preserving permissions, ownership, etc                     |
| <b>-r</b>         | recurses subdirectories without preserving permissions, etc.                        |
| <b>--partial</b>  | continues partially downloaded files                                                |
| <b>--progress</b> | prints a progress bar while transferring                                            |
| <b>-P</b>         | is the same as <b>--partial --progress</b>                                          |

See the **rsync(1)** man page for a complete list

## End of Unit 7

- Questions and Answers
- Summary
  - Filesystem quotas
  - Configuration of Software RAID
  - Software RAID recovery
  - Configuration of Logical Volumes
  - LVM Snapshots
  - Backup Tools



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[training@redhat.com](mailto:training@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.











# Lab 7

## Advanced Filesystem Mangement

---

Goal: Develop skills and knowlege related to Software RAID, LVM, quota and backup.

Estimated Duration: 120 minutes

## Sequence 1: Implementing Quotas

**Deliverable:** A user diskhog that cannot use more than 1024k of space in /home.

**Instructions:**

- ✓ 1. Create a user diskhog
- ✓ 2. Activate user quotas for /home
- ✓ 3. Set the soft block quota of user diskhog to 512 1k blocks and the hard limit to 1024 1k blocks.
- ✓ 4. Test the restrictions.

## Sequence 2: Software RAID

Deliverable: A system with two RAID1 devices with two partitions each.

### Instructions:

1. ✓ This lab assumes that the computers are only equipped with one harddrive. Setting up RAID on a single disk obviously does not provide any kind of fault protection.  
✓ Create four additional partitions with at least 100MB each.
2. ✓ Make sure that the kernel uses the new partition table:
3. ✓ Use **mdadm** to create two RAID 1 devices, with two partitions each. These devices should be `/dev/md0` and `/dev/md1`. Create a filesystem on the first device.
4. ✓ Mount the filesystem under `/data` and put some files on it.
5. ✓ Get a printout of the current RAID configuration.
6. ✓ It is now time to simulate a disk failure. Use **mdadm --fail** to mark one of your partitions as faulty. Use then **mdadm --remove** to remove this partitions from the RAID array. Check `/proc/mdstat` and `/var/log/messages` to see how the system reacts on this error.
7. ✓ If this had been a real hard disk error, you would have to replace the broken disk and repartition it. Since our partition was never really damaged we can skip this part and re-add the partition to RAID device with **mdadm --add**. After you re-add the partition, note the content of `/proc/mdstat`



## Sequence 3: Creating Logical Volumes with LVM

**Deliverable:** A volume group that uses one of the RAID1 devices as physical volume and has one logical volume.

### Instructions:

1. ✓ Unmount the filesystem from the last exercise and convert your first RAID device (`/dev/md0`) into a physical volume.
2. ✓ Create a volume group named `volgroup` with the default extent size of 4MiB, which uses the physical volume that was just created.
3. ✓ Create a logical volume named `data`, that uses all available extents. Create a filesystem on it and make sure that this filesystem is mounted to `/data` during boot.

✓ *Note:* LVM caches device information. Due to this cache new volume groups are ignored during boot. Delete `/etc/lvm/.cache`, so that LVM rescans the system for volume groups.

## Sequence 4: Extending a filesystem

**Deliverable:** A LVM that uses both RAID1 devices as physical volumes and has one increased logical volume.

### Instructions:

1. ✓ Add the second RAID device to your volume group.
2. ✓ Increase the ~~opt~~ logical volume and filesystem by 40MiB

*data*

## Challenge Sequence 5: Reducing a filesystem

**Deliverable:** The existing logical volume is reduced by 100MiB.

**Instructions:**

1. To reduce the volume the filesystem has to be reduced first. Reduce the filesystem by 100MiB.



## Challenge Sequence 6: Backing a Logical Volume with Snapshots

**Deliverable:** Take a backup from an active Logical Volume by using an LVM snapshot and dump. Destroy the data on the Logical Volume and recover from the backup.

**System Setup:** Confirm that the Logical Volume `/dev/volgroup/data` from an earlier lab is mounted on `/data` and configured in `/etc/fstab`

### Instructions:

1. Copy the contents of the `/var/log` directory into `/data`. View the contents of `/data`.
2. Create a read-only Snapshot Volume of `/dev/volgroup/data` called `data-backup`, allocate 16MB of space to the Snapshot.
3. Mount `/dev/volgroup/data-backup` read-only as `/mnt/data-backup` and view it's contents.
4. Use `dump` to backup `/mnt/data-backup` to a file named `/tmp/data-backup.dump`. Confirm that the backup file was created.
5. Unmount `/mnt/data-backup` and remove the snapshot volume.
6. Unmount `/data` create a new filesystem on `/dev/volgroup/data`. Remount `/data` and confirm that it is empty.
7. Use **dump** to recover the lost contents of `/data` from the backup. Check the contents of the directory.

## Sequence 1 Solutions

1. Create a user diskhog

```
useradd diskhog
```

2. Activate user quotas for /home

- a. In `/etc/fstab` change the mount options for `/home` to include `usrquota`. If you do not have a separate `/home` partition use `/`.

- b. Activate the new mount option.

```
mount -o remount /home
```

- c. Bring the system into single user mode, to ensure correct quota calculation:

```
init 1
```

- d. Run `quotacheck -cu /home` to create the quota file.

- e. Leave single user mode

```
init 5
```

- f. Enable Quota enforcing.

```
quotaon /home
```

3. Set the soft block quota of user diskhog to 512 1k blocks and the hard limit to 1024 1k blocks.

```
setquota -u diskhog 512 1024 0 0 /home
```

4. Test the restrictions.

To test these restrictions, run the following commands:

```
su - diskhog
quota
dd if=/dev/zero of=bigfile bs=1k count=400
quota
```

(Should work fine.)

```
dd if=/dev/zero of=bigfile bs=1k count=800
quota
```

(Should issue a warning.)

```
dd if=/dev/zero of=bigfile bs=1k count=1600
quota
```

(Should fail to write the whole file.)



## Sequence 2 Solutions

1. Create four additional partitions with at least 100MB each. The partition type should be set to "Linux raid auto".
  - a. Use **fdisk** to create four logical partitions
  - b. Set the Partition Type (**T**) to **fd**
  - c. Save and exit fdisk
2. Make sure that the kernel uses the new partition table:

```
partprobe
```

3. Use **mdadm** to create two RAID 1 devices, with two partitions each. These devices should be `/dev/md0` and `/dev/md1`. Create a filesystem on the first device.

*Note:* Adjust the partition numbers to match your system.

- a.

```
mdadm -C /dev/md0 -a yes -l 1 -n 2 ✓
/dev/sda{6,7}
mdadm -C /dev/md1 -a yes -l 1 -n 2 ✓
/dev/sda{8,9}
```

- b.

```
mkfs.ext3 /dev/md0
```

4. Mount the filesystem and put some files on it.

- ```
# mkdir /data
```
- ```
mount /dev/md0 /data
```

*Note:* Do NOT put it in `fstab` yet.

- ```
# cp -a /lib /data
```

5. Get a printout of the current RAID configuration.

```
# mdadm --detail /dev/md0
```

6. It is now time to simulate a disk failure. Use **mdadm --fail** to mark one of your partitions as faulty. Use then **mdadm --remove** to remove this partitions from the RAID array. Check `/proc/mdstat` and `/var/log/messages` to see how the system reacts on this error.

Note: The device names in this example might differ from your setup

a.

```
# mdadm --fail /dev/md0 /dev/sda7
```

b.

```
# cat /proc/mdstat
```

c.

```
# mdadm --remove /dev/md0 /dev/sda7
```

7. Read the partition to RAID device with **mdadm --add**.

a.

```
# mdadm -a /dev/md0 /dev/sda7
```

b.

```
# cat /proc/mdstat
```

Sequence 3 Solutions

1. Convert your first RAID device into a physical volume.

```
# umount /dev/md0
# pvcreate /dev/md0
```

2. Create a volume group named `volgroup` with the default extent size of 4MiB, which uses the physical volume that was just created.

```
# vgcreate volgroup /dev/md0
```

3. Create a logical volume named `data`, that uses all available extents. Create a filesystem on it and make sure that this filesystem is mounted to `/data` during boot.

Note: LVM caches device information. Due to this cache new volume groups are ignored during boot. Delete `/etc/lvm/.cache`, so that LVM rescans the system for volume groups.

- a. Use **`vgdisplay`** to find out how many extents are available.

```
# vgdisplay volgroup
```

- b. Create the logical volume with all available extents.

*Replace the word "free_extents" below with the number of free extents you found when running **`vgdisplay`** not the word "free_extents"*

```
# lvcreate -n data -l free_extents volgroup
```

- c.

```
# mkfs.ext3 /dev/volgroup/data
```

- d. Edit `/etc/fstab` so that `/data` will be mounted automatically.

```
/dev/volgroup/data    /data                ext3                  defaults 1 3
```

- e. Run **`mount -a`** to mount and check for errors.

- f. Delete the LVM devices cache to ensure that the new volume group is detected during boot.

```
# rm /etc/lvm/.cache
```


- g. Reboot to verify that the filesystem is automatically mounted.

Sequence 4 Solutions

1. Add the second RAID device to your volume group.

a.

```
# pvcreate /dev/md1
```

b.

```
# vgextend volgroup /dev/md1
```

2. Increase the opt logical volume and filesystem by 40MiB. Verify the increased size.

a.

```
# lvextend -L +40M /dev/volgroup/opt
```

b.

```
# resize2fs /dev/volgroup/opt
```

c.

```
# df -h
```

Challenge Sequence 5 Solutions

1. To reduce the volume the filesystem has to be reduced first. Reduce the filesystem by 100MiB.
 - a. First you need to determine the current filesystem size. This can be done with `df -h`
 - b. The filesystem has to be umounted before reducing

```
# umount /data
```

- c. Make sure that the filesystem is in a consistent state before reducing.

```
# fsck -f /dev/volgroup/data
```

- d. Now reduce the filesystem by 100MiB. This example assumes that the original size was 140MiB.

```
# resize2fs /dev/volgroup/data 40M
```

- e. It is now possible to reduce the logical volume. Be careful with this command. Wrong parameters can render your filesystem unusable.

```
# lvreduce /dev/volgroup/data -L 40M
```

- f. Verify the new size, by mounting the filesystem and running `df -h`

Challenge Sequence 6 Solutions

1. Copy the contents of the `/var/log` directory into `/data`. View the contents of `/data`.

a.

```
# cp -r /var/log/*  
/data/
```

b.

```
# ls /data/
```

2. Create a read-only Snapshot Volume of `/dev/volgroup/data` called `data-backup`, allocate 16MB of space to the Snapshot.

```
# lvcreate -L 16M -p r -s -n data-backup  
/dev/volgroup/data
```

3. Mount `/dev/volgroup/data-backup` read-only as `/mnt/data-backup` and view its contents.

a.

```
# mkdir /mnt/data-backup
```

b.

```
# mount -o ro  
/dev/volgroup/data-backup /mnt/data-backup
```

4. Use `dump` to backup `/mnt/data-backup` to a file named `/tmp/opt-backup.dump`. Confirm that the backup file was created.

a.

```
# dump -0u -f  
/tmp/data-backup.dump /mnt/data-backup
```

b.

```
# ls -la /tmp/data-backup.dump
```

5. Unmount `/mnt/data-backup` and remove the snapshot volume.

a.

```
# umount  
/mnt/data-backup
```

b. `# lvremove /dev/volgroup/data-backup`

6. Unmount /data create a new filesystem on /dev/volgroup/data. Remount /data and confirm that it is empty.

a.

`# umount /data`

b.

`# mkfs.ext3 /dev/volgroup/data`

c.

`# mount /data`

d.

`# ls /data`

7. Use **dump** to recover the lost contents of /data from the backup. Check the contents of the directory.

a.

`# cd /data`

b.

`# restore -rf
/tmp/data-backup.dump`

c.

`# ls`

Unit 8

Network Configuration



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Objectives

Upon completion of this unit, you should be able to:

- configure IP interfaces
- setup routes
- understand name resolution
- setup IPv6



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Network Interfaces

- Networking scripts refer to logical interface names:
 - Ethernet: eth0, eth1 ...
 - Dial-up: ppp0, ppp1 ...
 - Loopback: lo *localhost*
- Display network interfaces by using:
 - **ifconfig -a**
 - **ip link [show]**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Logical Interface Naming

The Linux kernel names interfaces with a specific prefix depending on the type of interface. For example, all Ethernet interfaces start with eth, regardless of the specific hardware vendor. Following the prefix, each interface is numbered, starting at zero. For example, eth0, eth1, and eth2 would refer to the first, second, and third Ethernet interfaces.

Additionally, it is possible to assign multiple Layer 3 addresses to a single physical network adapter. In this case, the second, third, and so on layer 3 addresses would be assigned to individual “device aliases”. The naming of a “device alias” takes the logical adapter name appending a colon and a unique alias number: eth0:1, eth0:2, and eth0:3 representing the second, third and fourth layer 3 addresses.

Display Hardware Configuration

The hardware address of network interfaces can be determined by running the **/sbin/ifconfig** command. Without options, **ifconfig** will only display the “active” interfaces. To view “all” interfaces, including “inactive” ones, use the **-a** option.

```
[root@stationX ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:9A:86:9D
          inet addr:172.31.53.128  Bcast:172.31.53.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe9a:869d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31 errors:0 dropped:0 overruns:0 frame:0
          TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3726 (3.6 KiB)  TX bytes:7310 (7.1 KiB)
          Interrupt:185 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2935 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2935 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:0  
RX bytes:3908056 (3.7 MiB) TX bytes:3908056 (3.7 MiB)
```

Another method to view “all” of your interfaces and their hardware addresses is to use **/sbin/ip link [show]**.

```
[root@stationX ~]# ip link  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 00:0c:29:9a:86:9d brd ff:ff:ff:ff:ff:ff  
3: sit0: <NOARP> mtu 1480 qdisc noop  
    link/sit 0.0.0.0 brd 0.0.0.0
```


Driver Selection

- All drivers for network interface cards are built as modules
- `/etc/modprobe.conf` maps logical names to specific modules:
 - `alias eth0 3c59x`
- Secondary “card selection” can be specified in the interface configuration file, `/etc/sysconfig/network-scripts/ifcfg-eth0`
 - `HWADDR=00:0D:60:FB:CA:61`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Driver Association

Red Hat compiles network card drivers as kernel modules for easy adaptability to any hardware configuration. Network interface modules are loaded at boot time if networking has been enabled. The appropriate module is loaded based on an alias line in `/etc/modprobe.conf`.

```
[root@stationX ~]# grep 'alias eth' /etc/modprobe.conf
alias eth0 3c59x
alias eth1 e1000
alias eth2 e1000
```

Card Selection

If there is only one card, or, more specifically, only one card utilizing that module/driver, then the alias line is sufficient to associate the physical card with a logical adapter name, `ethX`.

In the above example, where there is more than one network card utilizing the same module, `e1000`, the mapping of specific card to logical adapter name will utilize the hardware address of the card. This is specified in the logical adapter's interface configuration file: `/etc/sysconfig/network-scripts/ifcfg-ethX` using the `HWADDR` variable:

```
[root@stationX ~]# grep 'HWADDR' /etc/sysconfig/network-scripts/ifcfg-eth*
/etc/sysconfig/network-scripts/ifcfg-eth0:HWADDR=00:0D:60:FB:CA:61
/etc/sysconfig/network-scripts/ifcfg-eth1:HWADDR=00:16:41:4C:84:55
/etc/sysconfig/network-scripts/ifcfg-eth2:HWADDR=00:16:41:59:AB:F5
```

Speed and Duplex Settings

- Modules are configured to autonegotiate, by default
- Mismatches can cause intermittent to no communication
- Manually overridden using:
 - **ethtool**
 - `ETHTOOL_OPTS` in `ifcfg-ethX`
 - `options` or `install` in `/etc/modprobe.conf` for older interface modules



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Speed and Duplex

Network cards can negotiate speed and duplex settings with the “hub/switch” that they are attached to. Autonegotiation allows the network card and upstream “hub/switch” to transparently select the optimal setting.

When there is a mismatch in speed and/or duplex between the settings presumed by the card and the settings presumed by the upstream “hub/switch”, communication becomes intermittent, at best, or non-existent. A mismatch will also exhibit error statistics of “overruns” or “frame/collsns” when viewing the interface through `/sbin/ifconfig` or `/sbin/ip -s link`

Manual Override

The speed and duplex settings for the card can be viewed and/or changed with `/sbin/ethtool`.

```
[root@stationX ~]# ethtool eth0
Settings for eth0:
    Supported ports: [ TP ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
    Supports auto-negotiation: Yes
    Advertised link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
    Advertised auto-negotiation: Yes
    Speed: 100Mb/s
    Duplex: Full
    Port: Twisted Pair
    PHYAD: 0
    Transceiver: internal
    Auto-negotiation: on
    Supports Wake-on: umbg
    Wake-on: g
    Current message level: 0x00000007 (7)
    Link detected: yes
```

Manual changes are best made when the interface is not “in use”. It is also recommended to turn off autonegotiation before “forcing” a manual setting. To manually force 100Mbps, full duplex operation on `eth1`:

```
[root@stationX ~]# ifdown eth1
[root@stationX ~]# ethtool -s eth1 autoneg off speed 100 duplex full
[root@stationX ~]# ifup eth1
```

To make the changes persist across a reboot, we need to incorporate the settings into the logical adapter interface configuration file, `ifcfg-ethX`, by adding the following `ETHTOOL_OPTS` line:

```
ETHTOOL_OPTS="autoneg off speed 100 duplex full"
```

Note: If you are working with an older module that does not support the **ethtool** interface, you may need to turn to either specifying module parameters with an `options` line or executing **/sbin/mii-tool** via an `install` line, both in `/etc/modprobe.conf`.

IPv4 Addresses

- View configuration with:
 - `ifconfig`
 - `ip addr [show]`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

To see the IP addresses assigned to each interface we can use either `/sbin/ifconfig` or `/sbin/ip addr [show]`.

```
[root@stationX ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:9a:86:9d brd ff:ff:ff:ff:ff:ff
    inet 172.31.53.128/24 brd 172.31.53.255 scope global eth0
    inet6 fe80::20c:29ff:fe9a:869d/64 scope link
        valid_lft forever preferred_lft forever
3: sit0: <NOARP> mtu 1480 qdisc noop
    link/sit 0.0.0.0 brd 0.0.0.0
```

Dynamic IPv4 Configuration

- Interface configuration defined in:
 - `/etc/sysconfig/network-scripts/ifcfg-ethX`
 - Dynamic with line of: `BOOTPROTO=dhcp`
- Zero Configuration Networking
 - Uses `169.254.0.0/16`
 - Disabled with line of: `NOZEROCONF=yes` in `/etc/sysconfig/network-scripts/ifcfg-ethX`
- Use ***ifdown device***; ***ifup device*** to apply configuration changes



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Dynamic IPv4 Configuration

Red Hat Enterprise Linux stores network interface configuration information in files in the `/etc/sysconfig/network-scripts/` directory. The file names are prefixed with `ifcfg-` followed by the logical adapter name. For example, the first Ethernet interface would be `ifcfg-eth0`.

DHCP (Dynamic Host Configuration Protocol) can be used to automatically obtain an IP address and other configuration parameters from a central server. The `BOOTPROTO` variable in the interface configuration file controls the use of ***dhclient*** to negotiate the lease:

```
[root@stationX ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=dhcp
HWADDR=00:0D:60:FB:CA:61
ONBOOT=yes
```

Zeroconf

If there is no DHCP server an address from the `169.254.0.0/16` network is automatically assigned. These addresses are non-routable.

The use of this IP address range can be disabled by adding a `NOZEROCONF=yes` to the interface configuration file, `/etc/sysconfig/network-scripts/ifcfg-ethX`.

Activating Configuration Changes

After making changes to the interface configuration file, the process of bringing up a network interface involves the potential interaction of several utilities and processes. For example, a dialup interface would need to instruct the ***pppd*** daemon to dial the modem, or static routes might need to be added, or, as configured above, the ***dhclient*** daemon needs to negotiate a lease from a DHCP server. The `/sbin/ifup` and `/sbin/ifdown` scripts take care of all the extra tasks that need to be performed when activating and deactivating a network interface.

```
[root@stationX ~]# ifdown eth0
```

```
[root@stationX ~]# ifup eth0
```

```
Determining IP information for eth0... done.
```


Static IPv4 Configuration

- Interface configuration defined in:
 - `/etc/sysconfig/network-scripts/ifcfg-ethX`
- Static with lines of:
 - `BOOTPROTO=none`
 - `IPADDR=10.0.0.1`
 - `NETMASK=255.255.255.0`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

When `BOOTPROTO` is set to either `static` or `none`, the **dhclient** daemon is not utilized and the network settings are additionally read from the interface configuration file. Here is an example of an interface that is activated at boot time with a static address:

```
[root@stationX ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
HWADDR=00:16:41:4C:84:55
BOOTPROTO=none
IPADDR=172.31.53.1
NETMASK=255.255.255.0
ONBOOT=yes
```

Note: Do not forget to activate your changes with `/sbin/ifdown` and `/sbin/ifup` after editing the interface configuration file.

Device Aliases

- Useful for virtual hosting
- Bind multiple IP addresses to a single NIC
 - `eth1:1`
 - `eth1:2`
 - `eth1:3`
- Create a separate interface configuration file for each device alias:
 - `ifcfg-ethX:xxx`
 - Must use static networking



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A common configuration is to “host” multiple web or ftp sites on a single server, often called “virtual hosting”. Separate IP addresses are generally required for each web site that supports SSL or when defining multiple ftp sites.

Device aliases are virtual network interfaces that can be configured with an additional Layer 3 network address. The name of the device alias takes the logical adapter name appending a colon and unique number representing the alias: *if-name:alias-value*. For example, an alias of the `eth1` interface might be named `eth1:1`.

When using device aliases, the additional IP address must be defined statically. Create an additional interface configuration file named for the device alias, i.e. `ifcfg-if-name:alias`, as displayed below:

```
[root@stationX ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth1:1
DEVICE=eth1:1
BOOTPROTO=none
IPADDR=10.53.0.1
NETMASK=255.255.255.0
ONBOOT=yes
```

Routing Table

- Defines path to all systems
- View table with:
 - **route**
 - **netstat -r**
 - **ip route [show]**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The routing table allows the system to transmit packets to remote systems. **/sbin/route**, **/bin/netstat -r**, and **/sbin/ip route [show]** can all be used to list the current routing table:

```
[root@stationX ~]# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
172.31.53.0      *               255.255.255.0    U        0      0      0 eth0
default          172.31.53.2     0.0.0.0          UG        0      0      0 eth0

[root@stationX ~]# netstat -r
Kernel IP routing table
Destination      Gateway          Genmask          Flags  MSS Window  irtt Iface
172.31.53.0      *               255.255.255.0    U        0 0      0 eth0
default          172.31.53.2     0.0.0.0          UG        0 0      0 eth0

[root@stationX ~]# ip route
172.31.53.0/24 dev eth0 proto kernel scope link src 172.31.53.128
default via 172.31.53.2 dev eth0
```

When a packet is received by the kernel for transmission, the destination IP address of the packet is compared against each of the “Destination” networks specified in the routing table. On closest match, the packet is physically transmitted to the “Gateway” of the matching entry.

Using the **ip route** output from above, a packet destined for 172.31.53.176 would have a closest match to:

```
172.31.53.0/24 dev eth0 proto kernel scope link src 172.31.53.128
```

In this case, the destination is “locally” attached to this system (note the lack of **via**) and the packet would be sent physically to the destination IP address of the packet, out the device **eth0**.

Alternatively, a packet destined for 192.168.53.42, would match “default”, an alias for the network “0.0.0.0”:

```
default via 172.31.53.2 dev eth0
```

The destination is “remote” (note the use of **via**) and the packet would be sent physically to the router at 172.31.53.2, again out the device **eth0**.

Default Gateway

- Used when no route entry is matched
- Might be obtained dynamically with DHCP
- Can be statically configured:
 - With a line of: `GATEWAY=10.53.0.254`
 - Globally in: `/etc/sysconfig/network`
 - OR, per interface in the interface configuration file:
`/etc/sysconfig/network-scripts/ifcfg-ethX`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Defining the Default Gateway

The default gateway (or router) specifies where IP packets should be sent where there is no “more specific match” found in the routing table. It is also generally used when there is only “one way out” of the local network(s).

If you are using DHCP to configure your NIC, the DHCP server is usually configured to serve an address of the default gateway. The **dhclient** daemon will retrieve that value and set the default gateway in the routing table.

If you have configured your network cards statically, you have a choice of where to define the default gateway. The most common place is in the global network configuration file, `/etc/sysconfig/network`. Simply add or edit the `GATEWAY` line to manually declare the default gateway:

```
[root@stationX ~]# cat /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=localhost.localdomain
GATEWAY=10.53.0.254
```

Alternatively, you can associate a default route with a specific interface, by placing the `GATEWAY` variable in the interface configuration file: `/etc/sysconfig/network-scripts/ifcfg-ethX`. When such an interface is activated, that default route will override any previous default gateway. As an example, if the `GATEWAY` variable is defined in both the global network configuration file and the interface configuration file, the one in the interface will “win”.

Configuring Routes

- To control traffic flow when there is more than one router
- Static routes defined per interface
 - `/etc/sysconfig/network-scripts/route-ethX`
 - Uses **ip route add** syntax
- Dynamic routes learned via daemon(s)
 - **quagga**
 - Support for various forms of RIP, OSPF, and BGP



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Additional Routes

The routing table will contain entries for each of the directly attached networks, automatically. When there is only one router, it is usually defined to be the “Default Gateway/Router”. When there is more than one router, or more than one interface each attached to different routers, we may want to selectively control which traffic goes through which router by configuring additional routes.

Static Routes

Additional routes may be configured statically on a per-interface basis. Routes may be added manually using the `/sbin/ip route add network/length via gateway` command:

```
[root@stationX ~]# ip route add 192.168.22.0/24 via 10.53.0.253
```

will add a static route, sending all traffic destined for the 192.168.22.* network physically through the locally connected 10.53.0.253 router. There must be a network interface currently active and directly attached to the network that the 10.53.0.253 router resides on.

To make these static routes persist across a reboot, the `/sbin/ifup` and `/sbin/ifdown` scripts will read from `/etc/sysconfig/network-scripts/route-ethX`. Routers referenced in the file of that interface would be locally attached to the network of that interface. Lines in this file use the same syntax as the **ip route add** command:

```
[root@stationX ~]# cat /etc/sysconfig/network-scripts/route-eth1
192.168.22.0/24 via 10.53.0.253
```

Dynamic Routing Protocols

One of the challenges in managing static routes is what to do when the network “changes”. Routers use dynamic routing protocols to inform each other of these “changes”, allowing the routers to re-calculate a potentially new “shortest path”. Router Information Protocol (RIP) is generally used in smaller networks. Open Shortest Path First (OSPF) is a protocol that supports larger enterprises. Internet Service Providers and larger hosted sites might employ Border Gateway Protocol (BGP) to assess preferred routes between various backbones on the Internet.

Our Linux server can participate in these “router” conversations and ultimately update the local routing table dynamically as the network changes. Support for these dynamic routing protocols can be found in the package **quagga**.

Verify IP Connectivity

- **ping**
 - Network packet loss and latency measurement tool
- **traceroute**
 - Displays network path to a destination
- **mtr**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Verify Connectivity

/bin/ping attempts to test network connectivity by sending ICMP (Internet Control Message Protocol) packets to a specific system on the network. If the remote system is accessible via the network, it will reply. Failure to receive a reply could indicate a problem with the local host, the intervening routers, or even the remote host. Generally, we should check more than one remote host to verify connectivity.

The default behavior of **ping** is to send a 64-byte ICMP packet to the specified host every second until you cancel the operation with a **Ctrl-C**. When the ping operation is canceled or done, **ping** will report summary statistics such as average packet loss, number of packets sent/received, etc. These statistics can allow us to measure the “quality” of the network between hosts.

The **-c #** option allows us to specify a count of the number of ICMP packets **ping** should send before simply ending.

```
[root@stationX ~]# ping -c 4 172.31.53.2
PING 172.31.53.2 (172.31.53.2) 56(84) bytes of data.
64 bytes from 172.31.53.2: icmp_seq=1 ttl=128 time=0.396 ms
64 bytes from 172.31.53.2: icmp_seq=2 ttl=128 time=0.411 ms
64 bytes from 172.31.53.2: icmp_seq=3 ttl=128 time=0.399 ms
64 bytes from 172.31.53.2: icmp_seq=4 ttl=128 time=0.289 ms

--- 172.31.53.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.289/0.373/0.411/0.054 ms
```

Learn the Path

As network traffic travels across the Internet, it usually crosses through multiple routers. When connectivity between the local system and a remote system is sluggish and/or inconsistent, it is useful to investigate which router is responsible for the network problem.

The **/bin/traceroute** command will attempt to show the path of routers that network packets take between the local system and a remote system. It uses a gradually escalating TTL (Time To Live) value to obtain an ICMP **TIME_EXCEEDED** packet from each router along the path. **traceroute** uses UDP frames by default to probe the path.

The **/usr/sbin/mtr** command combines the functionality of **ping** and **traceroute** into a single network diagnostic tool. See the **mtr(8)** man page for further details.

Defining the Local Host Name

- View/Set local hostname with **hostname**
- Initially defined in `/etc/sysconfig/network`:
 - `HOSTNAME=stationX.example.com`
- Might “pull” name from network
 - **dhclient** daemon
 - “Reverse DNS Lookup”



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Viewing the Local Host Name

While each system will have a unique Layer 3 address, most users would prefer to identify a given system by a “name”, rather than memorizing a “number”. We can view the name of a system by utilizing the `/bin/hostname` command.

```
[root@stationX ~]# hostname
stationX.example.com
```

Setting the Local Host Name

We can utilize the same **hostname** command to override and manually specify a hostname.

```
[root@stationX ~]# hostname localhost.localdomain
[root@stationX ~]# hostname
localhost.localdomain
```

The `/etc/rc.d/rc.sysinit` startup script will initially set the hostname to the `HOSTNAME` variable found in: `/etc/sysconfig/network`:

```
[root@stationX ~]# grep 'HOSTNAME' /etc/sysconfig/network
HOSTNAME=stationX.example.com
```

However, the `/etc/rc.d/init.d/network` startup script will attempt to “override” the above definition if it is missing or set to `localhost` or `localhost.localdomain`. It might do this in one of two ways. First, if the interface is set to dynamic, it will request a hostname from the DHCP server. Second, it will do a reverse lookup of the interface IP address, utilizing DNS (provided it is not the loopback adapter, a PPP interface, nor a SLIP interface). Once the name has been “changed”, the name would no longer be undefined or equal to some form of `localhost`. Consequently, if the first interface is successful, subsequent interfaces will not be used to “change” the name.

Local Resolver

DNS

- Resolver performs forward and reverse lookups
- `/etc/hosts`
 - Local database of hostname to IP address mappings
 - Useful for small isolated networks
 - Normally, checked before DNS



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Using Names

Since users will want to refer to the systems by name, the system needs to convert those names into corresponding IP addresses. A “resolver” looks up the number when we have the name (forward lookup), or looks up the name when we have a number (reverse lookup).

Local Resolver

If you have a small network of computers, you can eliminate DNS lookups for communication between hosts by creating a standard `/etc/hosts` file and copying it on to all of your machines. Of course, every time you want to make a change, you need to replicate that change out to all of your machines. This does not scale very well.

However, at a minimum, your `/etc/hosts` must contain an entry for `localhost`, and, ideally, would contain an entry referring to the hostname you had defined in `/etc/sysconfig/network`. Do not adjust the `localhost` lines, other than to maybe “add” an alias to the “end” of that line.

```
[root@stationX ~]# cat /etc/hosts
127.0.0.1      localhost.localdomain  localhost
::1           localhost6.localdomain6 localhost6
172.31.53.1    stationX.example.com  stationX
```


Remote Resolvers

- `/etc/resolv.conf`
 - Domains to search
 - Strict order of name servers to use
 - May be updated by **dhclient**
- `/etc/nsswitch.conf`
 - Precedence of DNS versus `/etc/hosts`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

DNS Domains

To allow for distributed administration of the names defined on the Internet, names follow a general convention of `hostname.domain`, placing them in a tree-like hierarchy, known as DNS (Domain Name System). When looking for a host, one can specify either the “fully-qualified domain name (FQDN)” or an abbreviated form of the name, in effect leaving off the “domain” portion of the name. FQDNs officially end with a period to specify that they are, in fact, FQDNs.

The `search` or `domain` line in `/etc/resolv.conf` allows the definition of domain names that can be automatically searched when the user has not “fully-qualified” the host name they are attempting to resolve.

Name Servers

A Name Server represents a centralized repository of hostnames and Layer 3 addresses, usually for a particular domain. There are several methods of having name servers transparently query other name servers (and hence domains) for additional records.

We define a list of name servers to be checked on each individual system. The `nameserver` lines in `/etc/resolv.conf` are checked in order. The first name server listed will be queried, and, if it is unavailable, the second name server will be queried, and so on down the list.

```
[root@stationX ~]# cat /etc/resolv.conf
search example.com another.org
nameserver 172.31.35.2
nameserver 192.168.1.3
```

dig = X

While this file can be managed individually on each system, the **dhclient** daemon will automatically obtain a list of name servers from the DHCP server unless the interface configuration file contains: `PEERDNS=no`

The client resolver will search either DNS or the local `/etc/hosts` file for an answer based on the `hosts` line in `/etc/nsswitch.conf`.

Verify DNS Connectivity

- Verify name servers using:
 - **nslookup** (deprecated)
 - **host**
 - **dig**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `bind-utils` package provides several utilities to query name servers:

- **host** is a simple tool that allows forward and reverse lookups.

```
[root@stationX ~]# host server1.example.com
server1.example.com has address 192.168.0.254
[root@stationX ~]# host 192.168.0.254
254.0.168.192.in-addr.arpa domain name pointer server1.example.com.
```

- **dig**, can be used to provide a standardized, parseable, and detailed output. It does only work with fully qualified host names.
- **nslookup** is deprecated and may be removed in future releases.

Network Configuration Utilities

- **system-config-network**
 - **system-config-network-gui**
 - **system-config-network-tui**
- Profile Selection
 - **system-config-network-cmd**
 - `netprofile` kernel argument



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Network Configuration Utilities

Through the Gnome or KDE panels, we can navigate to System->Administration->Network which will launch **system-config-network-gui**. From the command-line, we can enter **system-config-network** which will launch either the GUI version (**system-config-network-gui**) or the text based version (**system-config-network-tui**) depending on the environment the command has been executed from.

system-config-network provides a friendly method of modifying the files we have referred to throughout this unit. You can setup Ethernet, ISDN, PPP, xDSL, Token Ring, CIPE, and/or wireless network interfaces for static or dynamic configuration. You can define routes, hostnames, name servers, etc.

Profiles

The **system-config-network** framework, however, maintains a parallel directory structure under `/etc/sysconfig/networking/`. This opens up another feature, allowing the creation of *profiles*, the definition of multiple, alternative network configurations. When we transition from one profile to another, the files under `/etc/sysconfig/networking/` are *dynamically linked* to the normal network configuration files.

We can further automate the transition from one networking profile to another by using the **system-config-network-cmd** command directly or via a script.

```
[root@stationX ~]# system-config-network-cmd --profile ProfileName --activate
```

At boot time, we can also pass a network profile via a kernel argument perhaps as an alternative stanza in `/boot/grub/grub.conf`.

```
title RHEL5 with ProfileName networking
    root (hd0,0)
    kernel /vmlinuz-version ro root=LABEL=/ netprofile=ProfileName
    initrd /initrd-version.img
```

The use of profiles can simplify the network configuration of a system “on the move”. Usually applied to laptops, you could, for example, have one network profile for Work and another for Home. Create two entries in `/boot/grub/grub.conf` to boot your system with the appropriate network settings depending on where you are.

Transparent Dynamic Configuration

- **NetworkManager** service
- **nm-applet**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Too Many Profiles

The list of *profiles* managed through **system-config-network** can quickly get out of hand. For example, you could need one for your cube in the office, one for the conference room, one for your home office, yet another when you want to lounge in front of the TV, not to mention taking your laptop to the coffee shop on Main, the myriad of hotels your company sends you too, and on, and on. Your `/boot/grub/grub.conf` is getting rather large, not to mention that some of these transitions are not involving rebooting....

Transparent Dynamic Configuration

The **NetworkManager** service coupled with the user-space tool, **nm-applet** more fully integrate the configuration of the network with the graphical desktop environment, the focus on mobile computing.

As you disconnect your laptop from the “wired connection” in your cube on your way to the conference room, the **NetworkManager** service transparently switches you from your wired configuration over to your wireless adapter. When you take that first trip to the coffee shop on Main and boot your laptop, you interact with **nm-applet** to specify the wireless settings, wireless network and security key, which will be automatically remembered for your next trip.

There are a couple of caveats, though. It is currently presumed that the interfaces are dynamically configured (i.e., use DHCP, not static settings). Secondly, due to the interactive nature of the configuration, the interfaces are not activated until you first log in. Both of these issues will be explored/addressed in a future version of **NetworkManager**.

To enable **NetworkManager**, it is recommended that you disable all physical interfaces at boot time, so that the `/etc/rc.d/init.d/network` startup script does not activate them. Modify each of the `/etc/sysconfig/network-scripts/ifcfg-ethX` physical interface configuration files so that `ONBOOT=no`. Then, restart the **network** service and enable the **NetworkManager** service:

```
[root@stationX ~]# service network restart
[root@stationX ~]# chkconfig NetworkManager on
[root@stationX ~]# service NetworkManager start
```


Implementing IPv6

- Kernel **ipv6** module enables stateless autoconfiguration
- Additional configuration implemented by **/etc/rc.d/init.d/network** initialization script
 - `NETWORKING_IPV6=yes` in `/etc/sysconfig/network`
 - `IPV6INIT=yes` in `/etc/sysconfig/network-scripts/ifcfg-ethX`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Enabling/Disabling IPv6

IPv6 support begins with a kernel module, `ipv6.ko`. To disable IPv6, we must prevent the loading of the module by adding the following two lines to `/etc/modprobe.conf`:

```
alias net-pf-10 off
alias ipv6 off
```

Stateless Autoconfiguration

If the module is loaded, active interfaces will have the default address automatically assigned. This is similar to the Zeroconf facility used in IPv4. These addresses are locally-scoped, i.e. non-routable.

Additional configuration options are managed through the `/etc/rc.d/init.d/network` startup script. IP version 6 startup and configuration code will be executed if the `NETWORKING_IPV6=yes` setting is found in `/etc/sysconfig/network`.

Interface configuration of dynamic or static settings will be executed if the `/etc/sysconfig/network-scripts/ifcfg-ethX` contains `IPV6INIT=yes`.

Viewing Interface Settings

We can view the settings of the interface with `/sbin/ip [-6] addr [show]`.

```
[root@stationX ~]# ip -6 addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 fe80::20c:29ff:fe9a:869d/64 scope link
        valid_lft forever preferred_lft forever
```

IPv6: Dynamic Interface Configuration

- Two ways to dynamically configure IPv6 addresses:
 - Router Advertisement Daemon
 - Runs on (Linux) Default Gateway - **radvd**
 - Only specifies prefix and default gateway
 - Enabled with `IPV6_AUTOCONF=yes`
 - Interface ID automatically generated based on the MAC address of the system
 - DHCP version 6
 - **dhcpc6s** supports more configuration options
 - Enabled with `DHCPV6C=yes`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Stateful Autoconfiguration

Each IPv6 router can be configured to advertise itself and the “network” it is attached to. A Linux-based router can run the **radvd** daemon to make these advertisements. This advertisement is limited to only autoconfiguring the “network” portion of the IPv6 address coupled with a default gateway of the router making the advertisement.

Listening for and utilizing these advertisements is controlled through the variable `IPV6_AUTOCONF` defined, globally, in `/etc/sysconfig/network`, or, overridden on a per interface basis, in `/etc/sysconfig/network-scripts/ifcfg-ethX`.

The “learned” prefix is then combined with an automatically generated Interface ID using EUI-64 (a 64-bit derivation of the MAC-48 address). Due to the global and predictable nature of EUI-64, RFC 3041 was developed to protect our privacy. This is enabled by setting `IPV6_PRIVACY=rfc3041` in `/etc/sysconfig/network-scripts/ifcfg-ethX`.

DHCP version 6

The more traditional method of managing dynamic layer 3 addresses was met with Dynamic Host Configuration Protocol (DHCP). A new DHCP has been developed to support IP version 6. Unlike version 4, the new **dhcpc6s** daemon does not listen for “broadcasts”, but rather subscribes to the multicast address: `ff02::16`. But, as with version 4, a multitude of values can be delivered to the clients.

Setting `DHCPV6C=yes` in `/etc/sysconfig/network-scripts/ifcfg-ethX`, will enable the launching of the `/sbin/dhcpc6c` daemon to request dynamic configuration from the server.

IPv6: StaticInterface Configuration

- `/etc/sysconfig/network-scripts/ifcfg-ethX`
 - `IPV6ADDR=<ipv6_address>[/prefix_length]`
 - Device aliases unnecessary...
 - `IPV6ADDR_SECONDARIES=<ipv6_address>[/prefix_length] [...]`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

While automatic and/or dynamic configuration of client systems is useful, servers will normally want a fixed assignment.

Utilizing the interface configuration file, `/etc/sysconfig/network-scripts/ifcfg-ethX`, the first static Global Unicast address is defined using the `IPV6ADDR` variable.

```
[root@stationX ~]# grep 'IPV6ADDR' /etc/sysconfig/network-scripts/ifcfg-eth0
IPV6ADDR=2001:db8:100:0::1/64
IPV6ADDR_SECONDARIES=2001:db8:100:1::1/64 2001:db8:100:2::1/64
```

Unlike with IP version 4, additional Global Unicast addresses can be directly assigned to the interface. There is no need for device aliases, like `eth0:1`. Instead, specify a list of additional IPv6 Global Unicast addresses using the variable, `IPV6ADDR_SECONDARIES`.

Note: Do not forget to activate your changes with `/sbin/ifdown` and `/sbin/ifup` after editing the interface configuration file.

IPv6: Routing Configuration

- Default Gateway
 - Dynamically from **radvd** or **dhcpcv6s**
 - Manually specified in `/etc/sysconfig/network`
 - `IPV6_DEFAULTGW=<IPv6_address [%interface]>`
 - `IPV6_DEFAULTDEV=<interface>` - only valid on point-to-point interfaces
- Static Routes
 - Defined per interface in `/etc/sysconfig/network-scripts/route6-ethX`
 - Uses **ip -6 route add** syntax
 - `<ipv6_network/prefix> via <ipv6_routeraddress>`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Routing Configuration

The default gateway can be learned via one of the automated methods of Global Unicast address assignment. However, to manually define the default gateway, utilize the `IPV6_DEFAULTGW` variable in `/etc/sysconfig/network`.

```
[root@stationX ~]# cat /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=localhost.localdomain
GATEWAY=10.53.0.254
IPV6_DEFAULTGW=2001:db8:100:1::ffff
```

When there is more than one router to send to, and we want to control the flow/direction of selected traffic, we can declare which networks should be sent through which routers using static routes. These are defined on a per interface basis in `/etc/sysconfig/network-scripts/route6-ethX` utilizing the same syntax as the utility **ip -6 route add**.

```
[root@stationX ~]# cat /etc/sysconfig/network-scripts/route6-eth0
2001:db8:100:5::/64 via 2001:db8:100:1::ffff
```

New and Modified Utilities

- **ping6**
- **traceroute6**
- **tracepath6**
- **ip -6**
- **host -t AAAA *hostname6.domain6***



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

- **/bin/ping6** - test connectivity and measure response time
- **/bin/traceroute6** - display the routers crossed to a remote host
- **/bin/tracepath6** - document the routers to a remote host and discover the MTU along this path
- **/sbin/ip** - view and/or configure IPv4 and IPv6 routing and device settings
- **/usr/bin/host** - lookup IPv6 address of host via DNS

End of Unit 8

- Questions and Answers
- Summary
 - Where are drivers linked to specific interfaces?
 - Where is a static IP address defined?
 - Where is the default route set?
 - Where is the list of nameservers stored?



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Lab 8

Manage Network Settings

Goal:	To build skills needed to manually configure networking
Estimated Duration:	45 minutes
System Setup:	A Red Hat Enterprise Linux System using DHCP networking
Lab Setup:	Disable the DHCP service on server1.

Sequence 1: Setting up a static IP address

Scenario: The DHCP server is down! You need to get your workstation up on the network, so you have to edit the appropriate configuration files by hand to set up static networking.

Deliverable: A system configured to operate with static rather than dynamic network settings.

Instructions:

1. Current Settings

What is your current IP address and subnet mask

✓ it varies .48 . 255.1.1.0

What is your current default gateway

What is your current DNS Name Server

2. Shut Down Interface

Shut down your interface

3. Define Layer 3 Static Settings

✓ Edit the configuration file of the eth0 device, so that it uses the IP Address, Network Mask, and Default Gateway settings recorded above.

4. Verify DNS and Hostname Resolution

✓ Verify that your system is still using the recorded name server from above.

✓ Make sure that your hostname is set to stationX.example.com and that this hostname is used automatically at boot.

Verify that your hostname can be resolved to the correct IP address without using DNS.

5. Test new settings

✓ Bring up your newly-configured interface and verify that your settings work.

Make sure that this configuration also works after a reboot.

Station B

Sequence 2: Virtual IP Addresses and Static Routes

Scenario: A network service will be installed on your system. You have decided to use a virtual IP address for this service to be more flexible in the future.

Your coworker has likewise created a virtual IP address that is not routed by your gateway but which you need to gain access.

Deliverable: An additional virtual IP address available after reboot.

A static route to support access to virtual IP address of your neighbor.

System Setup: Work together with a partner. Your virtual IP address will be 10.0.X.1/24 (where X is your station number). The virtual IP address of your neighbor will be 10.0.Y.1/24 (where Y is the station number of your neighbor).

Instructions:

1. **Create IP alias**

Configure your system with a new device alias named `eth0:1` so that it uses the additional IP address 10.0.X.1 with a network mask of 255.255.255.0. This IP should also be available after reboot.

2. **Create Static Route to Neighbor Alias**

Create a static route to the alias network of your neighbor via the regular IP address of your neighbor:

```
network 10.0.Y.0/255.255.255.0
through gateway 192.168.0.Y
```

Make this route permanent so that it persists across reboots.

Sequence 3: Troubleshooting Slow DNS

Scenario: Our users are complaining about slow network response from this system with no additional details.

Instructions:

1. Break DNS

Modify the list of nameservers to contain a total of three: `192.168.0.X`, `192.168.1.X`, `192.168.0.254`, in that specific order, where *X* is your station number.

2. Symptoms

Try pinging `192.168.0.254`. It should be successful and the first ping should be almost immediate.

Try pinging the same machine, but using its name, `server1.example.com`. The operation should be successful, but should have paused for a couple of seconds before transmitting and reporting the first ping response.

This should lead us to a determination that something is wrong with name resolution.

3. Troubleshoot and Fix

Test DNS connectivity with `dig`. What DNS Server is responding with an answer

Compare the reported DNS server to the list of nameservers configured on this system. In what position is the reported name server

The client resolver uses the list of nameservers in a fixed sequence *every time*. Hence the delay in resolving since the first two entries are not “responding”.

Fix the list of nameservers by returning it back to its original state.

Sequence 1 Solutions

1. What is your current IP address and subnet mask

```
[root@stationX ~]# ip addr show
```

What is your current default gateway

```
[root@stationX ~]# ip route show
```

What is your current DNS Name Server

```
[root@stationX ~]# cat /etc/resolv.conf
```

2. Shut down your first ethernet interface

```
[root@stationX ~]# ifdown eth0
```

3. Open /etc/sysconfig/network-scripts/ifcfg-eth0 in a text editor and add or edit the following lines (where ~~#~~ is replaced with your station number). Leave other existing lines alone (like HWADDR):

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.0.X
NETMASK=255.255.255.0
GATEWAY=192.168.0.254
```

4. Verify that your system still uses 192.168.0.254 as nameserver. Adjust it if necessary.

```
[root@stationX ~]# cat /etc/resolv.conf
search example.com
nameserver 192.168.0.254
```

Make sure that your hostname is set to stationX.example.com and that this hostname is used automatically at boot.

```
[root@stationX ~]# hostname
stationX.example.com
```

Open /etc/sysconfig/network in a text editor and ensure that the HOSTNAME variable is properly set:

```
HOSTNAME=stationX.example.com
```

Verify that your hostname can be resolved to the correct IP address without using DNS. Open /etc/hosts in a text editor and ensure that the following lines appear:

```
127.0.0.1          localhost.localdomain  localhost
```

```
::1      localhost6.localdomain6 localhost6
192.168.0.X      stationX.example.com stationX
```

5. Bring up your newly-configured interface and verify that your settings work.

```
[root@stationX ~]# ifup eth0
[root@stationX ~]# ping server1
```

Make sure that this configuration also works after a reboot.

```
[root@stationX ~]# shutdown -r now
[root@stationX ~]# ping server1
```


Sequence 2 Solutions

1. Configure your system with a new device alias named `eth0:1` so that it uses the additional IP address `10.0.X.1` with a network mask of `255.255.255.0`. This IP should also be available after reboot.

- Create `/etc/sysconfig/network-scripts/ifcfg-eth0:1` in a text editor and enter the following lines (where `X` is your station number):

```
DEVICE=eth0:1
BOOTPROTO=none
ONBOOT=yes
IPADDR=10.0.X.1
NETMASK=255.255.255.0
```

- To verify, either reboot the system or simply activate the new device.

```
[root@stationX ~]# shutdown -r now
```

or

```
[root@stationX ~]# ifup eth0:1
```

- Test that the new interface responds locally.

```
[root@stationX ~]# ping 10.0.X.1
```

2. Create a static route to the alias network of your neighbor via the regular IP address of your neighbor:

```
network 10.0.Y.0/255.255.255.0
through gateway 192.168.0.Y
```

Make this route permanent so that it persists across reboots.

- Start by adding the following route manually:

```
network 10.0.Y.0/255.255.255.0
through gateway 192.168.0.Y
```

Note: If you have not used the **ip** command before, this is an excellent time to begin using it. Check `ip (8)` if you do not know the syntax.

```
[root@stationX ~]# ip route add 10.0.Y.0/24 via 192.168.0.Y
```

- Verify this new route.

View the current routing table to confirm the above line was added:

```
[root@stationX ~]# ip route
10.0.Y.0/24 via 192.168.0.Y dev eth0
```


See if you can access the remote network by trying to access the device alias of your neighbor:

```
[root@stationX ~]# ping -c 4 10.0.Y.1
```

- Make this route permanent and then restart networking to verify your changes persist.

Create a file named `/etc/sysconfig/network-scripts/route-eth0` with the following content:

```
10.0.Y.0/24 via 192.168.0.Y
```

Restart networking

```
[root@stationX ~]# service network restart
```

Test that remote alias is still available

```
[root@stationX ~]# ping -c 4 10.0.Y.1
```

Sequence 3 Solutions

1. Modify the list of nameservers to contain a total of three: 192.168.0.X, 192.168.1.X, 192.168.0.254, in that specific order, where X is your station number.

Open `/etc/resolv.conf` in a text editor and ensure that the following lines appear (in the same sequence, replacing X with your station number):

```
search example.com
nameserver 192.168.0.X
nameserver 192.168.1.X
nameserver 192.168.0.254
```

2. Try pinging 192.168.0.254. It should be successful and the first ping should be almost immediate.

```
[root@stationX ~]# ping -c 4 192.168.0.254
```

Try pinging the same machine, but using its name, `server1`. The operation should be successful, but should have paused for a couple of seconds before transmitting and reporting the first ping response.

```
[root@stationX ~]# ping -c 4 server1
```

This should lead us to a determination that something is wrong with name resolution.

3. Test DNS connectivity with `dig`. What DNS Server is responding with an answer

```
[root@stationX ~]# dig server1.example.com
```

Compare the reported DNS server to the list of nameservers configured on this system. In what position is the reported name server

```
[root@stationX ~]# cat /etc/resolv.conf
```

The client resolver uses the list of nameservers in a fixed sequence *every time*. Hence the delay in resolving since the first two entries are not “responding”.

Fix the list of nameservers by returning it back to its original state.

Open `/etc/resolv.conf` in a text editor and ensure that the following lines appear:

```
search example.com
nameserver 192.168.0.254
```




Unit 9

Installation



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Objectives

Upon completion of this unit, you should be able to:

- Know important command line switches
- Understand different installation methods
- Create advanced partition layouts
- Understand Kickstart's role
- Create Kickstart files



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Anaconda, the Red Hat Enterprise Linux Installer

- Supports different modes
 - Kickstart offers automated Installation
 - Upgrade performs an update of an existing Red Hat Enterprise Linux installation
 - Rescue Mode allows troubleshooting of unbootable systems
- Consists of two stages:
 - First stage starts the installation
 - Second stage performs the installation



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Anaconda is the installer for Red Hat Enterprise Linux. The Python based program features several modes. *Kickstart* mode provides an automated installation. This mode is discussed later in this unit.

The *Upgrade* mode can be selected when the Installer detects an existing Red Hat Enterprise Linux installation. It upgrades existing packages and leaves configuration and user data intact.

The *Rescue* mode can be used to troubleshoot a system that is no longer bootable. It loads the necessary rescue environment from the second stage and gives the user a shell to access the system. This mode is discussed in the Troubleshooting unit.

Anaconda is run in two stages. The first stage boots the system and performs initialization of the system. It will then load the second stage from the selected installation medium.

General information about the installation program are part of the **anaconda** RPM. If **anaconda** is installed, documentation can be found in `/usr/share/doc/anaconda-*/`. For additional information check the Red Hat Enterprise Linux Installation Guide on <http://www.redhat.com/docs>.

First Stage: Starting the Installation

- The first stage consists of a installation kernel and an `initrd.img`
- Can be started with any supported boot loader
- Tasks of the First Stage:
 - Initializes the Installer
 - Parses command line arguments
 - Autodetects hardware
 - Loads additional drivers
 - Selects language, keyboard layout and installation method
 - Sets up networking if required for installation



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The first stage loads Anaconda, the Red Hat Enterprise Linux installer. It consists of a special kernel and `initrd.img`. These files can be loaded by any supported boot loader. Images for alternative installation boot media can be found on the first installation disk or installation DVD.

The first stage parses command line arguments and then initializes the selected mode of the Installer. If booted from a installation CD it automatically starts the second stage. Otherwise it queries for language, keyboard layout and installation source. It sets up networking if the user selects a network based installation method. You can specify the **askmethod** command line argument to force a network installation from an installation CD.

Hardware is autodetected by the installer. This may hang the installation under rare conditions. In this case specify **noprobe** to disable autodetection.

Some hardware requires additional drivers to function. The hardware vendor can provide a driver disk to enable the device during installation. The installer automatically prompts for such a disk if autodetection fails or the **dd** command line argument is used.

First Stage: Boot Media

- Supported boot media:
 - `boot.iso` or Installation CD/DVD
 - USB drive containing `bootimg.img`
 - Network boot with PXE
 - Other bootloaders such as GRUB
 - Boot floppies no longer supported
- Boot media can modified for custom installations



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Boot CDs

The `boot.iso` file contains a small CD filesystem image that can be burned to CD with the **cdrecord** command:

```
cdrecord -v boot.iso
```

You can create your own `boot.iso` to customize the installation process

1. Put the contents of the `boot.iso` image into a temporary directory.

```
# mkdir /tmp/iso
# mount -o loop boot.iso /mnt
# cd /mnt
# cp -a . /tmp/iso
```

2. Put additional files (e.g. kickstart files) into the same directory.
3. Edit `isolinux/isolinux.cfg` to modify/add boot entries.
4. You can also modify `isolinux/boot.msg` to provide a custom boot message.
5. Create a new bootable image with the following commands:

```
# cd /tmp/iso
# mkisofs -o ../bootcd.iso -b isolinux/isolinux.bin -c \
isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -Jr .
```

USB Drives

The `diskboot.img` file contains a VFAT filesystem image designed to be written to a USB storage device with the **dd** or **cat** commands. For example, on an already installed linux system, a USB thumb-drive might be accessed as `/dev/sda1`. A boot image can be created on the USB device with the command:

```
cat diskboot.img > /dev/sda1
```

(WARNING! This will also destroy any data that was on the device)

Of course, this will only be useful if the BIOS supports booting from USB storage devices. Booting from a custom USB device may be preferred because it is the easiest way to customize the installation process.

Network Boot

PXE, the Pre-eXecution Environment, provides a method of automating your install such that no physical media needs to be inserted into the target system (if the BIOS supports it). Information on configuring PXE can be found in RHEL 4 System Administration Guide:

<http://www.redhat.com/docs/manuals/enterprise/>

Accessing the Installer

- Graphical Installation
 - Default installation type
 - Useful Switches: `lowres`, `resolution`, `skipddc`
- VNC based Installation
 - Activate with `vnc` and protect the session with `vncpassword=password`
 - Set network parameters with `ip=IP Address` and `netmask=Network Mask`
- Text based Installation
 - Started with the `text` switch
 - Menu-based terminal interface
- Serial Installation
 - Used automatically when no graphic card is detected
 - Enable with: `serial=device`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The graphical interface makes installation easy and intuitive. The graphical interface can be started in **lowres** mode, which means it uses lower screen resolution settings for the installation. The **resolution** switch can be used to specify a resolution such as 1024x768. If monitor detection does not work properly it can be disabled by using the **skipddc** switch.

The VNC interface is identical to the graphical interface. Once the second Stage of the installer is running you can access it with the **vncviewer**. Alternatively you can use the **vnconnect** switch, so that the installer connects to an listening vncviewer.

The text based installer is useful when the installer has difficulty managing your display adapter. While this is uncommon, it can be particularly useful on laptops that have proprietary display adapters. The text based installation does not support all installer features.

First Stage: Installation Method

- Available Installation Methods:
 - Local CDROM
 - Hard drive
 - NFS image
 - FTP
 - HTTP
- Media sets:
 - Two Sets available: Client and Server
 - Can be downloaded from Red Hat Network
 - May contain packages from additional layered products
 - An "Installation Key" must be entered to unlock additional content
 - Extra packages can also be installed after installation through RHN.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Installation Methods

Red Hat Enterprise Linux can be installed via several methods in addition to the traditional installation CDs. The user can place the ISO images of the media set onto EXT2 or VFAT filesystems on a partition of a harddisk, even external USB drives.

For NFS based installations the images or their contents must be located on a NFS share. FTP or HTTP installations require that the contents of the CDs are put into a public directory.

Media Sets

Red Hat Enterprise Linux is now delivered on two distinct media sets: Client and Server. These media sets can be downloaded from RHN at https://rhn.redhat.com/network/software/download_isos.pxt Media sets may also contain packages from certain layered products.

To avoid installation of packages which are not covered by the support contract, an Installation Key must be entered to enable these additional features. If the key is omitted only the base system is installed. The packages can also be obtained from RHN once the system is registered.

Network Installation Server

- Necessary for network-based installs
- Often faster than CDROM-based installation methods
- Provides an easy distribution platform for the enterprise
- Shares the RedHat directory via NFS, FTP and/or HTTP
- Can be used as a yum repository



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A network installation server can be easily created from a host running an NFS, FTP, and/or HTTP server.

To create an installation server, select one or more installation protocols:

If using anonymous FTP, copy each install CD's contents to the publicly-accessible FTP directory. It is safe to overwrite files such as TRANS.TBL.

cp -a /mnt/cdrom/. /var/ftp/pub

If using HTTP, either use the above technique to copy the contents to a location in your web file tree. If you choose to provide installation services via both FTP and HTTP, create a symbolic link from your web file tree to your existing FTP directory:

```
[root@stationX ~]# ln -s ../../ftp/pub /var/www/html/pub
```

```
[root@stationX ~]# chcon -h -R -t httpd_sys_content_t /var/ftp/pub
```

If using NFS, create an entry in /etc/exports to share the pub directory, for example:

```
/var/ftp/pub hosts.and.or.networks(ro)
```

Restart the servers you have chosen to configure, if necessary.

Second Stage: Installation Overview

- Language and keyboard selection
- Installation Key
- Disk partitioning
- Bootloader configuration
- Network and time zone configuration
- Package selection



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The second stage Installer first configures language and keyboard mapping if this was not been done by the first stage. It then requests an Installation Number to access additional packages on the installation media. If the Installation Key is skipped, only packages from the base distribution will be installed.

Then Partitioning and boot loader configuration is performed. You can also specify a boot loader password. After the Network and Time Zone screens package selection allows customization of the installed RPMs.

Configuring File Systems

- Must select mount points, partition sizes, and file system types in the installer
 - Can set up manually or automatically
- There are many layouts which may be used
 - / must include /etc, /lib, /bin, /sbin, /root
 - Swap space is typically 2x physical RAM
 - Typical mount points: /boot, /home, /usr, /var, /tmp, /usr/local, /opt



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Configuring the file system hierarchy

At installation time, you must divide up your disks into partitions of various sizes and identify whether those partitions should be formatted with a file system (typically ext3), used as swap space, or used as a RAID or LVM partition. If the partition contains a file system, it must also be assigned a mount point. You can have the installer automatically make these decisions, or you can make them manually.

If you choose automatic configuration, you still have some input in the partitioning process. You can ask to review and modify the selections manually after the installer makes its decisions. You can select which drives to use for the installation. You can also indicate if the installer should delete all existing partitions, delete all Linux partitions from previous installations, or leave all existing partitions alone (using unallocated disk space for new partitions).

You have a great deal of freedom in how you may manually configure your file system hierarchy. You must have a file system mounted on /. You typically should have about twice your RAM in swap space. The /etc/ /lib /bin and /sbin/ directories may not be on separate file systems; they must be part of the / file system or the system will not boot properly.

It is common to have a /boot file system about 100 MB in size at the front of the disk, to hold files needed by the BIOS at boot time (such as the kernel and parts of the boot loader). This helps to avoid problems with old BIOS code. One limitation on /boot is that most boot loaders expect it to be on a normal disk partition or RAID 1 device.

The /var directory holds files that change frequently. This includes log files, the mail spool, and temporary space for software updates from Red Hat Network. If /var is on a separate partition, it should probably be at least 1 GB in size.

Depending on how much software you choose to install, /usr will probably need between 1 GB and 5 GB of space. The /tmp directory should have enough free space to accommodate temporary files written by programs. The amount of space needed will vary depending on the number and type of applications and services running on the system. One gigabyte should be more than enough for most systems' /tmp directories. The rest of / requires a few hundred megabytes. This does not take into account any space needed for personal user files or software not included with RHEL.

Advanced Partitioning

- **Software RAID**
 - Create new partitions and select **Software RAID** as “filesystem” type
 - Combine RAID partitions into a RAID device with **RAID**
- **LVM**
 - Select **Physical Volume** to create physical volumes
 - **LVM** creates a Volume Group
 - **Add** creates new Logical Volumes



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

RAID

To create a RAID device first create partitions with the RAID “filesystem” type. Then select **RAID** to configure RAID level and partitions to use. Make sure that partitions forming the RAID device are located on different physical disks.

Anaconda automatically configures `/etc/mdadm.conf` to allow monitoring of the RAID devices. After installation edit the file to select another email address for email notifications.

LVM

A Volume Group is automatically created by the default partitioning. Select **LVM** to add more logical volumes. By default all available space is used. It might be advisable to leave some space unused in the Volume Group.

Limitations

The System's BIOS must be able to access the second stage of the boot loader in `/boot`. `/boot` cannot be placed on a RAID device except on RAID1. It can not be part of a Volume Group. It is therefore recommended to put `/boot` on a separate filesystem.

Package Selection

- A default set of packages is automatically installed
- Select **Customize now** to change the default set of packages
- Customizing is necessary to add support for additional languages
- Anaconda automatically resolves package dependencies
- Package set can easily customized after install with **yum** or **system-config-packages**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Anaconda installs a default set of packages unless you select **Customize now** during installation. This is required to add support for additional languages. It is easy to customize the package set after installation using **yum** or **system-config-packages**. Therefore it is advised to only install a minimal package set during installation.

First Boot: Post-Install Configuration

- Configure X Window System if necessary
- Firewall and SELinux Setup
- Kdump setup
- Set date and time
- Register with Red Hat Network and get updated RPMs
- Setup users
- Configure sound card
- Install additional RPMs or Red Hat documentation from CDROM



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Enterprise Linux runs a graphical program called firstboot if the system is booted into run-level 5 after the installation. firstboot offers several configuration functions such as setting the date and time, installing additional software, or registering for Red Hat Network.

Firewall Setup

The default firewall configuration blocks most incoming connection by default. You will be presented with two choices for the firewall, **Enabled** and **Disabled**.

Trusted Services allows you to let remote machines access particular services through the firewall. Some common services are listed. After the install is complete, you can use **system-config-securitylevel** to allow additional services in the **Other ports:** dialog box. This box takes a series of port:protocol pairs. For **port** you may use either the name from `/etc/services` or the port number; for example, both `imap` and `517` are acceptable definitions.

Firewall rules are written to `/etc/sysconfig/iptables` and invoked at boot up by the `/etc/rc.d/init.d/iptables` script.

SELinux

SELinux is automatically put into Enforcing mode. You can change the mode and configure booleans in the **Modify SELinux Policy** tab during first boot.

Kdump

Kdump can perform crash dumps. If a kernel panic occurs, the system switches to a second kernel image that then dumps key parts of the memory into swap space. For this some memory must be reserved.

Kickstart

- Scripted Installation method
- Supports all Anaconda features
- `/root/anaconda-ks.cfg` is automatically created during Install
- Configuration utility: **system-config-kickstart**
- Syntax-checker: **ksvalidator**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Kickstart allows the installer to read information from a designated file rather than prompting the person doing the installation for it. It especially facilitates the setup of a number of machines that have similar hardware that the installer can autoprobe successfully. If a required item is omitted from the Kickstart file, the installation pauses and the user is prompted for that information.

Anaconda automatically generates a Kickstart file during installation and saves it under `/root/anaconda-ks.cfg`. This file can be used as a basis for your own modifications.

system-config-kickstart is a graphical tool for creating and modifying Kickstart files. A separate syntax checker is also available: **ksvalidator**

Starting a Kickstart Installation

- Anaconda enters Kickstart mode, when the `ks` boot option is specified
- `ks` queries DHCP for the Kickstart location
- `ks=url` gets the file via HTTP, FTP, or NFS
- From a local medium: `ks=floppy`, `ks=cdrom`, or `ks=hd:device:/path/to/file`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Anaconda enters Kickstart mode when the `ks` option is given on the command line. Example:

```
boot: linux ks=http://server1/workstation.cfg
```

DHCP based Kickstart

A DHCP server simplifies network-based Kickstarts, even if the system to be built will use a static IP address. It may also instruct the Kickstarted client about the location of the Kickstart file and the server and NFS share on which it may be found. If there is no DHCP available network configuration must be entered manually.

If the location given is a file, then the client tries to mount the file's parent directory to retrieve the file. If the location given is a directory, then the client tries to mount that directory and looks for a file whose name is *DHCP-IP-Address-Kickstart*.

If the DHCP server does not provide a `next-server` name, then the client assumes the DHCP is the server on which the Kickstart file resides. If the DHCP server does not provide a filename, then the client assumes the directory is */Kickstart* and looks for a Kickstart file with a name of the *DHCP-IP-Address-Kickstart* form.

URL based Kickstart

It is also possible to specify a URL for the Kickstart file, like this:

```
linux ks=http://server/path/to/kickstart
```

This URL can point to a CGI script or other dynamic pages. If your Kickstart file is provided by NFS use this syntax:

```
linux ks=nfs:/server:file
```

Kickstart from Local Media

Kickstart files can also be placed on local media.

<code>ks=floppy</code>	<code>ks.cfg</code> located in the top level directory of a EXT2 or VFAT floppy
<code>ks=cdrom</code>	<code>ks.cfg</code> located in the top level directory of a CDROM
<code>ks=hd:device:file</code>	file is located on a VFAT or EXT2 filesystem.

Anatomy of a Kickstart File

- **Commands section**
 - Configures the system
 - Omitted directives are prompted to the user
- **%packages Section**
 - Selects packages and groups for installation
 - Dependencies are always resolved
- **Scripts section(s)**
 - Optional section to customize the system
 - %pre scripts are run before installation
 - %post scripts are run after installation



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A Kickstart file consists of several sections. In the first section the system is configured. If one of the mandatory directives is omitted, Anaconda will prompt the user. After that it continues with Kickstart.

The %packages section determines which packages are installed. The base system will always be installed, even if this section is empty.

At the end of the Kickstart file you can have optional %pre and %post scripts. These are used to customize the installation.

Kickstart: Commands Section

Starting the Installation

- Installation Mode
 - `install` performs a fresh install.
 - `upgrade` upgrades an existing installation.
- Installation Method:

```
cdrom
url --url url
nfs --server host --path directory
harddrive --partition=device --dir=/path/to/install_tree
```



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Kickstart: Commands section

Important Directives

- **Required Directives**
 - Must be specified, otherwise the installer configures them interactively
 - Localization options: `keyboard`, `lang`, `timezone`
 - Authentication: `rootpw`, `authconfig`
 - Bootloader: `bootloader`
- **Optional Directives**
 - Network: `network` [options]
 - Security: `firewall`, `selinux`, `services`
 - Installer behaviour: `firstboot`, `poweroff` | `reboot`, `interactive`, `text`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Required directives must be specified or the installer prompts the user. Optional directives are silently set to their defaults if omitted.

Required Directives

The `keyboard` directive sets keyboard mapping (e.g. `keyboard de-latin1` or `keyboard us`). The language is set with the `lang` directive. It takes the same arguments as the `LANG` environment variable. If set to something except `en_US` make sure to install the appropriate `@lang-support` package groups. `timezone` specifies the time zone of the system. Use the `--utc` option if your hardware clock is set to GMT.

The root password can be either be specified in clear text (`rootpw secret`) or crypted (`rootpw --iscrypted $1$Y2Qix4sC$dzS9MTJ4iscVkoVy9hNGu/`). Use the `authconfig` directive to control authentication. If no parameters are given the system uses local authentication and md5 password hashes.

The `bootloader` directive normally does not require arguments. Use `--password` or `--md5pass` to specify a bootloader password.

Security

Firewall settings can be controlled with the `firewall` directive. The `--disabled` or `--enabled` options determine if **iptables** should be activated. It is possible to specify with incoming network connections are allowed with the options `--ssh`, `--telnet`, `--http`, `--smtp`. Other ports can be specified with the `--port` option. This option takes comma separated pairs of `port:protocol`.

Selinux can be put into `--permissive` or `--enforcing` mode or completely `--disabled`.

Services can automatically started after installation. The `services` directive has two arguments: `--disabled` and `--enabled`. Both take a comma separated list of services.

Installer Behaviour

The `firstboot` directive specifies, if **firstboot** should be executed after installation. it can be `--enabled` (the default) `--disabled`. There is also a special `--reconfig` option, that cause firstboot to reconfigure language, keyboard and timezone.

You can specify if the system should be rebooted (`reboot`) or powered down (`poweroff`) after installation. If not specified the system prompts for reboot.

Kickstart can be turned into a interactive installation with default values with the `interactive` directive. You can run the Kickstart also in text mode with the `text` keyword.

Kickstart: Packages Section

- Add single packages with `package_name` without any version number
- Add package groups with `@package_group`
- Remove packages from the list: `-package_name`
- Use wildcards to specify multiple packages
- Dependencies are always resolved
- Add support for additional languages with `@lang-support`
- Packages from layered products can be installed when an installation key is specified by with the `key` directive in the `commands` section.



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `%packages` section adds packages to the installation. You can specify single packages, package groups (using `@package_group`) or remove packages included by a group with `-package_name`. Note that there is no version information, so the package section will also work with other releases with little changes.

Package Groups

Package groups are defined in `/path/to/install-media/Install-Class/repodata`. `Install-Class` is `Server` or `Client` for the core product. If an installation key is specified additional install classes are available.

It is no longer possible to specify `@Everything` for an all-inclusive installation. If you really want to install all available packages, use the new wildcard feature:

```
%packages
*
```

Support for additional languages can be added by installing `language-name-support` package groups. e.g

```
%packages
@german-support
```


Kickstart: %pre, %post

- %pre gives you the first word
 - executes as a bash shell script
 - executes after Kickstart file is parsed
- %post gives you the final word
 - Can specify interpreter (bash is default)
 - chrooted by default, but may be run without chroot



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The %pre and %post sections make just about anything possible. The %pre section executes as a bash script after the Kickstart file is read, but before partitioning, formatting, and copying of packages commences. The %pre script runs in a somewhat limited environment, as the only executables available are the ones provided by the installer. Unlike %pre, all of the packages specified in the %packages section are available in the %post section, which means many more utilities and capabilities are available.

The default behavior of %post is to execute the content of the section as a bash shell script in an environment that chroots to /mnt/sysimage -- the newly installed system. In other words, all paths and commands are as they will be on the installed system. The default use of bash as the interpreter may be overridden using the --interpreter switch on the %post line, e.g., %post --interpreter /usr/bin/perl. A non-chrooted environment is also possible through the --nochroot switch.

Examples:

```
%pre
# Create partitions by copying an MBR image
mknod /tmp/hda
dd if=/mnt/source/pub/mbr.img of=/tmp/hda

%post
# Download a customized xorg.conf file via ftp -- the echo allows a
# DHCP client system to use hostnames in the %post section
echo nameserver 192.168.0.254 >> /etc/resolv.conf

lynx -source ftp://server1/pub/xorg.conf > /etc/X11/xorg.conf

# Suppress the rewriting of /etc/resolv.conf on a DHCP client
cat >> /etc/sysconfig/network-scripts/ifcfg-eth0 <<EOF
PEERDNS=no
EOF
```


End of Unit 9

- Questions and Answers
- Summary
 - Steps of the installation
 - Important Anaconda switches
 - **system-config-kickstart**
 - **ksvalidator**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



Lab 9

Installation and System-Initialization

Goal: Successfully install Red Hat Enterprise Linux.

System Setup: A computer capable of booting from CD.

Warning: If your system cannot boot from the provided `boot.iso` CD, you might need to change the boot order in the computer's BIOS; however, please do not change BIOS settings without being instructed to do so.

Sequence 1: Installing Red Hat Enterprise Linux

Scenario: A new server was just delivered. Before moving it into production, you plan to perform a test installation.

Deliverable: A basic system with a Volume Group which is mirrored

Instructions:

1. Perform an NFS-based install, using the following installation information:

Server: **192.168.0.254**

Mount point: **/var/ftp/pub**

Use DHCP for your own network configuration. Use only IPv4 networking. Configure keyboard mapping and language as you prefer. Skip the installation key.

2. Partition the disk according to the following table:

/boot 128 MB
2 * RAID 15000 MB

Remember that RAID is not a mount point; it is a filesystem type.

Note: If your hard disk is empty, you may receive a warning box. If asked to initialize the drive, Click Yes.

3. Combine the two RAID partitions into a RAID 1 device with “physical volume” as the filesystem type.
4. Create a volume group name “MyGroup” with the following logical volumes:

Name	Size	Mount Point
root	10000 MB	/
home	512 MB	/home
swap	512 MB	N/A

5. Use the default Bootloader, Network, and Security Settings. Choose “redhat” as the root password.
6. Install **system-config-kickstart** in addition to the default set of packages. This package is in **Base System/Administration Tools**. Do not add support for extra tasks. Add language packages if required. Monitor the progress of the installation on the other consoles. After reboot complete the initial setup, but do not register with Red Hat Network. Check the log files for errors.

Sequence 2: Kickstart Installation

Deliverable: A system that is automatically installed with kickstart.

System Setup: **system-config-kickstart** reads the list of available packages from RHN or yum repositories. Since the system is not currently connected to Red Hat Network, we must configure to use a yum repository on `server1`.

Put the following lines in `/etc/yum.repos.d/base.repo`:

```
[base]
name=Red Hat Enterprise Linux $releasever - $basearch
baseurl=http://server1.example.com/pub/Server
enabled=1
gpgcheck=0
```

Instructions:

1. Open `/root/anaconda-ks.cfg` in **system-config-kickstart**. Make sure that `server1:/var/ftp/pub` is used as **Installation Method**.

2. Partition the disk according to the following table:

```
/boot 128 MB
/      4096 MB
swap   512 MB
/home  1024 MB
```

linux ks = http://server1.example.com/pub/kickstart3.x

3. Select **Firewall Configuration**, and disable the default firewall. Leave SELinux active. Save the file and exit **system-config-kickstart**.

4. Use a text editor to add the line "`key --skip`" to the top of the kickstart file, just below the URL line, to avoid being prompted for the activation key. Retrieve student's encrypted password. Just below the partitioning information, add the following lines:

```
user --name=visitor --password=password
user --name=student --password=paste-password-here --iscrypted
```

5. Save the file, and check the contents by running it through **ksvalidator** to make sure that no syntax errors are in the file.

6. Make the kickstart file available: Several methods exist to make the kickstart file available for the installer. Which method to use depends on the hardware in your classroom; your instructor will inform you about this step.

7. Reboot your system using the boot media provided by the instructor. After the boot: prompt appears, use the command line provided by your instructor to start your kickstart installation. (The command line to use depends on which method was used to make the kickstart file available in the previous step, and may vary from class to class.)

Note: Specify `noipv6` as a boot option to deactivate IPv6 during installation.

If anything is missing from the kickstart file, the installer will raise a dialog allowing you to add in the required information. You will use this installation for the remainder of this course.

Sequence 1 Solutions

1. Perform an NFS-based install.
 - a. Bootup system using CD
 - b. Press *Enter* at the boot: prompt.
 - c. Choose the appropriate language.
 - d. Press *Enter* on the OK prompt.
 - e. Choose the appropriate keyboard.
 - f. Press *Enter* on the OK prompt.
 - g. Choose **NFS image** for the installation method
 - h. Configure TCP/IP. Select **Use dynamic IP configuration (BOOTP/DHCP)**. Deselect **Enable IPv6 support**.
 - i. Press *Enter* on the OK prompt.
 - j. Enter the appropriate information for an NFS installation

NFS Server Name: **192.168.0.254**
Red Hat Enterprise Linux directory: **/var/ftp/pub**
 - k. At this point Anaconda (the installer) will retrieve the necessary installation image and will probe the system for its monitor and mouse type and will finally present you with the welcome screen. Click **Next**.
 - l. Anaconda then checks if Red Hat Enterprise Linux is already installed on this system. If yes, it will offer an Upgrade instead of a fresh installation. Choose **Install Red Hat Enterprise Linux** Click **Next**.
 - m. Select **Skip entering Installation Number**. Click **Next**. Confirm by clicking **Skip**.
2. Partition the disk according to the following table:

/boot	128 MB
2* software RAID	15000 MB

Remember that software RAID is not a mount point; it is a filesystem type.

Note: If your hard disk is empty, you may receive a warning box. If asked to initialize the drive, Click Yes.

 - a. Select **Create custom layout**. Click **Next**.

- b. To remove existing partitions, select the partition to delete and the click **Delete Partition**.

Hint: If you mark an entire disk you can delete multiple partitions at once. You must remove existing LVM volumes or RAID meta devices before deleting partitions.

- c. Click **New**.
- d. Enter `/boot` as Mount Point. Use a fixed size of 128 MB. The filesystem type should be ext3 (default). Click **OK**.
- e. Create the two RAID partitions. Leave the Mount Point empty and select RAID as the filesystem type and again a fixed size of 8000 MB.

- 3. Combine the two RAID partitions into a RAID 1 device with “physical volume (LVM)” as the filesystem type.

- a. Click **RAID**
- b. Select Create a RAID device
- c. Set the RAID level to RAID1.
- d. Select the filesystem type physical volume (LVM)

- 4. Create a volume group named “MyGroup” with the following logical volumes:

Name	Size	Mount Point
root	10000 MB	/
home	512 MB	/home
swap	512 MB	N/A

- a. Click **LVM**
- b. Set the Volume Group name to “MyGroup”
- c. Click **Add** to add a new logical volume
- d. Select the **Mount Point** /
- e. Enter “root” as the **Logical Volume Name**
- f. Set the **Size (MB)** to 10000
- g. Click **OK** and repeat for the other volumes

- 5. Use the default Bootloader, Network, and Security Settings. Choose the appropriate locality settings and “redhat” as the root password.

- a. Use the default Boot Loader settings unless the instructor advises otherwise; do not create a Boot Loader password.
- b. Choose DHCP for networking and activate on boot
- c. Set the time zone as appropriate for your location; implement UTC if the instructor suggests it
- d. Set the root password to `redhat` (it is not a good password, but please use it anyway).

6. Install **system-config-kickstart** in addition to the default set of packages. This package is in **Base System/Administration Tools**. Do not add support for extra tasks. Add language packages if required. Monitor the progress of the installation on the other consoles. After reboot complete the initial setup, but do not register with Red Hat Network. Check the log files for errors.
- a. Select **Customize now** and click **Next**.
 - b. Select the **Base System** tab. Highlight **Administration Tools** and click **Optional packages**. Select **system-config-kickstart**.
 - c. Add language packages if required and click **Next**.
 - d. You should now be at the About to Install screen. Click **Next** to begin.
 - e. Track the progress of filesystem formatting by switching to `tty5` (**Ctrl-Alt-F5** will take you there; **Ctrl-Alt-F6** will return you to the installer)
 - f. After the reboot following the installation, complete the initial set up tool. Create a user account of your choice. Do not register the machine with Red Hat Network. Select **No, I prefer to register at a later time** followed by **No thanks, I'll connect later**.
 - g. Choose enable Firewall, Leave SELinux at the default state Active.
 - h. Once you have completed the installation and the newly-installed system has booted, log in as root and examine the following:
 - `/var/log/messages`
 - `/var/log/dmesg`

Sequence 2 Solutions

1. Open `/root/anaconda-ks.cfg` in **system-config-kickstart**. Make sure that `server1:/var/ftp/pub` is used as **Installation Method**.
 - a. Start **system-config-kickstart**.
 - b. Go to **File/Open File** and open the file `/root/anaconda-ks.cfg`
 - c. Select **Installation Method** and choose NFS as method. Set the **NFS Server** to `server1.example.com` and the **NFS Directory** to `/var/ftp/pub`.
2. Partition the disk according to the following table:

<code>/boot</code>	128 MB
<code>/</code>	4096 MB
<code>swap</code>	512 MB
<code>/home</code>	1024 MB

 - a. Go to **Partition Information**.
 - b. Select **Remove existing Linux partitions**.
 - c. Click on **Add**
 - d. Select `/boot` as **Mount Point**. Choose 128 MB as the **Size**.
 - e. Make sure that **Format partition** is selected.
 - f. Repeat with appropriate changes for the `/` file system and for the swap partition.
3. Select **Firewall Configuration**, and disable the default firewall. Leave SELinux active. Save the file and exit **system-config-kickstart**.
4. To avoid being prompted for an installation key during the kickstart, you may open `/root/ks.cfg` in a text editor and manually add the following line to the top of the file:

```
key --skip
```

This will limit the set of packages available to kickstart to the base repository for Server, however.

In another window, retrieve student's encrypted password.

```
# grep student /etc/shadow
student:$1$7WrNp1$NcInkNf7m5z6vSFMiBS611:13578:0...
```

Just below the partitioning information, add the following lines:

```
user --name=visitor --password=password
user --name=student --password=paste-password-here --iscrypted
```


5. Save the file, and check the contents by running it through **ksvalidator** to make sure that no syntax errors are in the file.

Run the command:

```
#ksvalidator /root/ks.cfg.
```

If there is no output produced by the command, the kickstart file has successfully validated. This means the syntax of the file is correct, but kickstart may still fail due to logical problems in the file.

6. Make the kickstart file available: Several methods exist to make the kickstart file available for the installer. Which method to use depends on the hardware in your classroom; your instructor will inform you about this step.
7. Reboot your system using the boot media provided by the instructor. After the boot: prompt appears, use the command line provided by your instructor to start your kickstart installation. (The command line to use depends on which method was used to make the kickstart file available in the previous step, and may vary from class to class.)

Note: Specify `noipv6` as a boot option to deactivate IPv6 during installation.

If anything is missing from the kickstart file, the installer will raise a dialog allowing you to add in the required information. You will use this installation for the remainder of this course.



Unit 10

Virtualization with Xen



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Objectives

Upon completion of this unit, you should be able to:

- Define Virtualization
- Understand Xen Terminology
- Xen Tools



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Virtualization with Xen

- Advantages of Virtualization
 - Effective resource usage
 - Managability
 - Security
- Key Concepts of Xen
 - Small Hypervisor
 - First “Domain” manages the system
 - supports full and para virtualization



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The goal of virtualization is usually more effective use of resources. Projects may require a dedicated operating system installation, but not an entire dedicated computer.

Virtualization has many additional advantages as well. It can greatly simplify provisioning, while adding flexibility at the same time. Advanced uses of Xen support features including moving a live running instance of an operating system from one physical computer to another. Such a live migration typically incurs no more than a 100ms delay.

Individual virtual machines are completely separated by the hypervisor. Thus they can only communicate over the network, if permitted. An intrusion in one domain does not affect other virtual machines.

Concepts

Hypervisor:

The Hypervisor is the manager of the Xen environment. It acts as a traffic cop for all virtualized operating systems, controlling and providing access to resources such as storage, CPU, and memory. Additionally, the Hypervisor controls migration, starting, stopping, and pausing of virtualized operating systems. It is started by the bootloader.

Domain:

Domain is the Xen term for the virtual machine in which a virtualized operating system runs. In the world of Xen, a domain does not exist until it is created, usually to start up, or install, a virtualized operating system. Domain-0 is a privileged domain that controls the hypervisor. All other domains are called “Domain-U”.

Virtualization Types

Paravirtualization:

Paravirtualization is the native mode of Xen in which virtualized operating systems include support for the Xen environment, as if it were a variety of x86 (or x86_64) CPU. The main advantage of this approach is performance. Virtualized operating systems running in this mode typically incur no more than a 5% performance impact vs bare hardware.

Full Virtualization:

In this mode, Xen provides a complete machine simulation to run operating systems which do not include Xen support. The extra operations needed to “trick” the virtualized operation system into believing that it is running on the bare hardware add overhead. As such, the performance impact of full virtualization is greater than with Paravirtualization. Additionally, this type of complete machine simulation requires hardware support.

Hardware Considerations

- **Minimum Requirements:**
 - Processor with PAE support
 - 256 MiB RAM per domain
 - 6 GiB hard drive per domain
- **Additional Considerations:**
 - CPU with VT/SVM for Full Virtualization
 - Shared Storage for Live Migration
 - Actual Storage needs will vary by application



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

CPU Requirements

Xen requires CPUs that support PAE (Physical Address Extension) to run. Most Desktop and Server CPUs support this for years. One notable exception are older Mobile CPUs.

Additionally, CPUs with either Intel's VT technology or AMD's SVM technology allow operating systems with no native Xen support to run on Xen in "Full" Virtualization mode.

Storage

While not a minimum requirement to run Xen, shared storage allows for the most flexible configuration options and management scenarios. For example, Live Migration (the relocating of a running Domain-U from one physical host to another) is only available with shared storage.

Preparing Domain-0

- Install the Domain0 as normal
- Boot the Xen hypervisor
- Start the **xend** management daemon



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The first domain started in a Xen environment is a special management domain called Domain-0. Domain-0 has special privileges to manage the hardware devices. It is also the only domain from which an administrator may send commands to the Hypervisor.

Installing a Xen environment is as easy as installing Red Hat Enterprise Linux 5. However, when you install a system which is going to run Xen, you must keep in mind that you are actually installing the operating system that is going to run in Domain-0, the management domain of your Xen environment.

Domain-0 must include the utilities needed to create, manage, and destroy domains. These will be included if you select Virtualization support as part of your initial operating system install. The kernel package and management software may also be installed via yum or rpm after the initial operating system installation.

Make sure that the hypervisor is booted by default. It will boot the Domain-0 kernel as a module.

Without the **Xend** daemon running you cannot view or even start Domain-Us. Note that although Domain-0 is automatically created when Xen starts, the operating system running in Domain-0 must be configured to start **Xend** at boot. This is easily accomplished with **chkconfig**.

Virtual Resources

- CPU
 - uses VCPUs (Virtual CPUs)
 - need not map directly to real CPUs
- Storage
 - Block devices
 - Simple files
- Network Devices
 - Bridged or routed to Domain0
 - By default mapped to `xenbr0`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

CPUs

A VCPU is a virtual CPU as seen by an operating system running in a Domain-U. Each Domain-U may be configured to include 1 or more vcpus.

The number of vcpus in a Domain-U need not correspond directly to the number of CPUs in a system, but the relationship must be considered when configuring Domain-U's.

Storage

Xen can either use block devices or simple files as storage for the virtual machines. Files can grow on demand, but must be in `/var/lib/xen/images` due to SELinux restrictions.

Both are presented as virtual block devices within Domain-U and can be partitioned normally. By default they are mapped to `xvda`

Network

Each Domain-U has a virtual network device that has a virtual crossover connection to a device in Domain-0. Domain-0 may provide bridged or routed networking to Domain-U's. By default all domains are connected to the same virtual bridge: `xenbr0`

Domain-U Configuration

- Defined Per Domain-U
- Virtual Block Devices
- CPUs
- Networking
- `/etc/xen/domain`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Any virtual machine other than Domain-0 is generically referred to as a Domain-U. Virtualized operating systems running in Domain-U have no visibility of the Hypervisor or the physical hardware underlying the Xen environment. Thus a Domain-U may be moved from one Xen environment to another without modification, even if the underlying physical hardware differs.

The configuration files for Domain-Us are located in the `/etc/xen/` directory of Domain-0. Each Domain-U that will run in the Xen environment has its own configuration file.

```
[root@stationX]#cat /etc/xen/example
# example domain-u config file
name = "server100"
memory = "256"
disk = [ 'file:/var/lib/xen/images/server100.img,hda,w' ]
vif = [ 'mac=00:16:3e:66:06:aa' ]
bootloader="/usr/bin/pygrub"
on_reboot    = 'restart'
on_crash     = 'restart'
```

For detailed information on the options available for Domain-U configuration files see `xmdomain.cfg(5)`. The config file can be automatically created with **virt-manager**.

Installing a new Domain-U

- **virt-manager**
 - Graphical frontend for managing domains
 - Provides a wizard for setting up new domains
 - Command line alternative: **xm**
- Define name of the domain
- Select storage type and number of CPUs
- Specify the location of the installer and optionally a kickstart file



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Virtual Machine Manager

Installation of the virt-manager rpm provides the Virtual Machine Manager, a powerful and easy to use graphical tool for creating and managing Domain-Us. Additionally, the Virtual Machine Manager supports kickstart urls to automate the install of Red Hat Enterprise Linux for Domain-Us.

The easiest way to setup new virtual machines is **virt-manager**. Select **File/New machine** and provide the necessary information. The installation tree is the same as for normal installations.

After you have entered all information the installer is automatically executed.

Note: Once the installation is completed the system must be started with the **xm** command.

Domain Management with xm

- Command line management tool
- Controlling domains
 - **xm <create|destroy>**
 - **xm <pause|unpause>**
 - **xm <save|restore> filename**
 - **xm <shutdown|reboot>**
- Monitoring
 - **xm list**
 - **xentop**
 - **xen console**



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The primary tool for managing Xen domains is the **xm** command line tool.

The **xm** tool sends commands to **Xend** which relays the commands to the Hypervisor. It is important to note that most commands are executed by the Hypervisor asynchronously.

xm create [-c] domain

The **create** command is used to start up a domain. Use the **-c** switch to be connected to the domain's console as it starts.

Using the **pause** and **unpause** commands you may suspend a domain in memory and resume it, respectively.

The **destroy** command immediately kills a domain with force. It is the equivalent of pulling the plug out of the guest domain.

Use the **shutdown** command to gracefully shutdown a domain, as if you had typed **shutdown -h now** on the console of the domain.

xm save domain /path/to/statefile saves a running domain to a state file so that it can be restored later. Once the **save** completes, the domain will no longer be running or consuming resources. This is like putting a computer in hibernation.

xm restore /path/to/statefile creates a domain from the named state file. This is similar to taking a computer out of hibernation.

xm list shows all running domains. **xentop** provides a top-like listing.

To access the local console of of Domain-U you can either use **virt-manager** or **xm console**. You can leave this terminal again with **Ctrl-]**

Activating Domains on boot

- **xendomains** Sys-V init script
- Starts/stops Domain-U's
- must link domain config files to `/etc/xen/auto`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

xendomains

The **xendomains** init script is used to have select Domain-U's automatically start when Domain-0 is booted. The script also takes care to shut down the Domain-U's when Domain-0 is shut down.

To have **xendomains** start/stop a Domain-U create a symbolic link to the Domain-U configuration file in `/etc/xen/auto/`.

```
[root@stationX] #cd /etc/xen/auto/  
[root@stationX] #ln -s ../example example
```


End of Unit 10

- Questions and Answers
- Summary
 - Xen Terminology
 - xm commands
 - xendomains



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



Lab 10

Exploring Virtualization

Goal: To explore the Xen virtualization environment and the creation of a Domain-U virtual machine.

Sequence 1: Installing the Xen Virtualization Environment

Deliverable: A Red Hat Enterprise Linux system running the Xen virtualization environment.

Instructions:

1. Using **yum**, install the packages needed to set up the Xen virtualization environment: `kernel-xen`, `xen`, and `virt-manager`.
2. Create an additional partition with at least 2000MB.
3. Edit the `grub.conf` file to make the xen kernel boot by default.
4. Reboot to the xen kernel.

Sequence 2: Creating a Domain-U Virtual Machine

Deliverable: A Domain-U virtual machine running Red Hat Enterprise Linux

Instructions:

1. Using **virt-manager** create a new virtual machine using the following configuration information:
 - System Name: `vm1`
 - Install Media URL: `ftp://server1/pub`
 - Kickstart URL: `ftp://server1/pub/gls/vm1.cfg`
 - VM Max Memory: 256 MB
 - VCPUS: 1
 - Normal Partition: `/dev/sda5`

Sequence 3: Stating and Managing Domain-Us

Scenario: It is now time to boot up the newly created virtual system by using the **xm create** command.

Deliverable: The new virtual system it up and running.

Instructions:

1. Use the **xm create** command to boot up the newly-created virtual system.
2. In another terminal window **ping** `vm1` domain and leave the ping running.

In your original terminal window **pause** and **unpause** the `vm1` domain. Note that it stops responding to pings when it is paused and starts as soon as it is unpause.

Sequence 1 Solutions

1. Using **yum**, install the packages needed to set up the Xen virtualization environment: `kernel-xen`, `xen`, and `virt-manager`.
 - a. **`yum -y kernel-xen xen virt-manager`**
2. Create an additional partition with at least 2000MB.
 - a. Use **`fdisk`** to create the logical partition
 - b. Save and exit `fdisk`
3. Edit the `grub.conf` file to make the xen kernel boot by default.
 - a. If the Xen kernel is the first kernel listed in `/boot/grub/grub.conf`, then edit that file to set `default=0`.
4. Reboot to the xen kernel. Verify that the kernel name has "xen" in it using the `uname` command.
 - a. **`uname -r`**

Sequence 2 Solutions

1. Using **virt-manager** create a new virtual machine using the following configuration information:

- System Name: `vm1`
- Install Media URL: `ftp://server1/pub`
- Kickstart URL: `ftp://server1/pub/gls/vm1.cfg`
- VM Max Memory: 256 MB
- VCPUS: 1
- Normal Partition: `/dev/sda5`

To create the new virtual machine, do the following:

- When the **Open Connection** dialog appears, select **Local Xen host** and click **Connect**.
- The **Virtual Machine Manager** window will open. Start the new virtual system wizard by selecting **New Machine...** from the **File>** menu. Click **Forward**.
- For **System Name** enter `vm1` and click **Forward**.
- For **Install Media URL** enter `ftp://server1/pub`. For **Kickstart URL** enter `ftp://server1/pub/gls/vm1.cfg`. Click **Forward**.
- Select **Normal Disk Partition**, enter `/dev/sda5` as the **Partition**. Click **Forward**.
- Set **VM Max Memory** and **Vm Startup Memory** to 256 MB and the **number of VCPUs** to 1. Click **Forward**.
- Review the summary screen and click **Finish** to boot and kickstart the new virtual machine. A window will open and a text kickstart install will run. When the installation finishes, select **Reboot** in the virtual machine's window.

Sequence 3 Solutions

1. Use the **xm create** command to boot up the newly-created virtual system.

- a. `[root@stationX]# xm create vm1`

2. In another terminal window **ping** the vm1 domain and leave the ping running.

In your original terminal window **pause** and **unpause** the vm1 domain. Note that it stops responding to pings when it is paused and starts as soon as it is unpaused.

- a. `[root@stationX]#xm pause vm1`

- b. `[root@stationX]#xm list`

Name	ID	Mem (MiB)		VCPUs
State Time(s)				
Domain-0	0	736	1 r-----	353.6
vm1	4	256	1 --p---	13.6

- c. `[root@stationX]#xm unpause vm1`



Unit 11

Troubleshooting



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Objectives

Upon completion of this unit, you should be able to:

- Develop a strategy for troubleshooting
- Fix problems in different areas of the Linux system
- Boot the system into various runlevels
- Use the Rescue environment

a start at lower layers



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Method of Fault Analysis

- Characterize the problem
- Reproduce the problem
- Find further information
- Eliminate possible causes
- Try the easy things first
- Backup config files before changing



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Begin by characterizing the problem. What are the symptoms? Is anything else "odd" about the system? Note suspicious log file entries and error messages.

Try to reproduce the problem. Does the problem still occur after the service is restarted? Rebooting the system is usually not necessary. If you suspect a problem in the start up sequence it is often enough to bring the system to single user mode and back up again.

Eliminate possible causes for the problem. For each possible reason for the error, find a way to confirm or refute the cause. Always start with the reason that is easiest to test.

Always backup the old configuration before changing it. This allows you to easily revert to the old config when a change does not solve the problem.

Fault Analysis: Gathering Data

- Useful commands
 - **history**
 - **grep**
 - **diff**
 - **find -cmin -60**
 - **strace *command***
 - **tail -f *logfile***
- Generate additional information
 - ***.debug** in syslog
 - **--debug** option in application



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

history displays the last 1000 commands of a user. With **grep** you can find strings in text files.

diff shows the difference of two text files. Thus you can easily see what directives have been changed since the last backup of the file.

find *start-dir* -cmin -60 lists all files that have been modified within the last 60 minutes.

strace can be used to gather further data from applications. The syntax is **strace *command*** when running the command, or **strace -p *PID*** when connecting to a running process.

tail -f can be used to follow log file as they are generated. This allows you to see the errors as they are being generated, which may shed some light on what the problem may be.

syslog can capture debug information from the kernel and from system services. The default setting in `/etc/syslog.conf` is to send all message of info or higher to `/var/log/messages`. To get all messages (debug and higher), add a line to the configuration file and restart the **syslog**

```
/etc/syslog.conf:  
*.debug /var/log/debug
```

You may also want to add an file to `/etc/logrotate.d/` to rotate that file, but more on that later.

Most applications have a debugging option (e.g., **--debug**, **-d**, etc.) that forces the program to run in debug mode, thus creating copious entries in your log files. These options are often configured using the files in `/etc/sysconfig/`

For instance, to run **xinetd** in debug mode, add this entry to `/etc/sysconfig/xinetd`, and restart the service:

```
EXTRAOPTIONS="-d"
```

```
[root@stationX]# service xinetd restart
```


Things to Check: X

- Never debug X while in runlevel 5!
- Try **system-config-display** first
- **X -probeonly**
- Is `/home` or `/tmp` full, or has the user reached a hard quota?
- Is **xfs** running?



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email ctraining@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

X can sometimes be problematic. The first thing to try when confronting X problems is **system-config-display**. If the X problem is occurring in runlevel 5, it could make logging in impossible in X or in virtual consoles. Reboot or change the system runlevel to runlevel 3.

Sometimes viewing the output of the X command itself is revealing. The **probeonly** switch will perform all tasks necessary to start the X server with out actually starting it. Thus, the start up messages are displayed. If the text scrolls off the screen, the material off screen can be viewed by holding down the Shift key and pressing the Page Up key (Shift/Page Down to scroll down again). The output may be viewed by holding down Shift and using Page Up and Page Down. `/usr/share/hwdata/Cards` may provide useful information about optional configuration features for your hardware.

A user's inability to create files in the user's home directory or in `/tmp` either because of a full filesystem or because of a hard quota limit typically inhibits the ability of the user to run X. Although the exact symptoms differ between runlevels, messages will appear stating (in runlevel 3) or suggesting (in runlevel 5) that a filesystem is full.

The xfs font server is the only font path entry in the `/etc/X11/xorg.conf` file, so if it is not running, X will not run. Make sure that xfs is configured to run in the appropriate runlevels. Occasionally it may be necessary to delete stale lock, pid, or socket files. Once in a while, the font indexes in a font directory may be corrupt, and it will be necessary to run **mkfontdir** to recreate them. When this happens, xfs may seem to start correctly, but then dies. Try commenting out font paths in `/etc/X11/fs/config`, then run xfs from a terminal to determine which directory has problems.

Remember that X is a network service, even when you have not enabled access to your display. Unsuccessful hostname resolution can produce various behaviors, including the inability to launch applications from the panel. Changing hostnames can cause similar problems; if you need to change your computer's hostname, switch out of runlevel 5 and make sure X is not running (through **startx**), change the hostname, and only then restart X.

Things to Check: Networking

- Hostname resolution
 - **dig** **www.redhat.com**
- IP configuration
 - **ifconfig**
- Default gateway
 - **route -n**
- Module specification
- Device activation



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Hostname resolution problems can create problems for clients and servers alike. Aside from requiring successful forward lookups, reverse lookups are essential for many host-based security mechanisms. Tools like **host** and **dig** are invaluable for determining whether hostname resolution problems exist.

IP configuration may be checked using the **ifconfig** command, which will print information such as an interface's IP, the subnet mask, and other important settings. The **netstat -r** and **netstat -rn** commands will show if a system's routing table is correct. The absence of a default gateway or the existence of multiple default gateways can create problems. Inability to contact the default gateway (and thus, to reach the gateway to get outside the local network) can also cause networking problems.

It is possible that the kernel module for your particular network interface card has been mis-specified. For example, the Red Hat Enterprise Linux installer sometimes probes a **de4x5**-based card as a **tulip**-based card. Unfortunately, the **tulip** module will only work enough to enable the interface, but not enough to work.

Do not overlook the obvious: maybe the interface has not been activated, or was deactivated for some reason.

Order of the Boot Process

- Bootloader configuration
- Kernel
- **/sbin/init**
 - Starting init
- **/etc/rc.d/rc.sysinit**
- **/etc/rc.d/rc**, **/etc/rc.d/rc?.d/**
 - Entering runlevel X
- **/etc/rc.d/rc.local**
- X



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

In order to troubleshoot boot time problems, one must understand the boot process itself, remember how things look when they are working correctly, and narrow down how far into the process a failure is occurring.

Issue: No bootloader splash screen or prompt appears *Possible Causes:* GRUB is misconfigured. Boot sector is corrupt. A BIOS setting, such as disk addressing scheme, has been modified since the boot sector was written.

Issue: Kernel does not load at all, or loads partially before a panic occurs *Possible Causes:* Corrupt kernel image. Incorrect parameters passed to the kernel by the bootloader.

Issue: Kernel loads completely, but panics or fails when it tries to mount root filesystem and run **/sbin/init** *Possible Causes:* Bootloader is misconfigured. **/sbin/init** is corrupted or **/etc/inittab** is misconfigured. Root filesystem is damaged and unmountable.

Issue: Kernel loads completely, and **/etc/rc.d/rc.sysinit** is started and interrupted *Possible Causes:* **/bin/bash** is missing or corrupted. **/etc/fstab** may have an error; evident when filesystems are mounted or fsck'ed. Errors in software RAID or quota specifications. Corrupted non-root filesystems (due to a failed fsck).

Issue: Run level errors (typically services) *Possible Causes:* Another service required by a failing service was not configured for a given runlevel. Service-specific configuration errors. Misconfigured X or related services in run level

5

Filesystem Corruption

- Common after crash or improper shutdown
- ext2 mounted for writing marked “dirty”
 - If not mounted or mounted read-only, “clean”
 - if not mounted and “dirty”, may be corrupted
 - repair requires exhaustive check
- ext3 usually marked “clean”
 - journal indicates if recovery is needed
 - only need to check files recorded in journal



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Filesystem corruption is one of the more common boot-time problems. It can occur after a system crash causes the machine to shut down without correctly unmounting its filesystems. Filesystem corruption happens because the operating system uses RAM as disk buffers to improve performance; when power fails, information written to memory buffers which is not yet synchronized to the disk is lost.

When a device using an traditional filesystem like ext2 is mounted read-write, the filesystem is marked on disk as dirty. Only when the umount command is issued and all data is safely written to the disk is the disk marked clean and unmounted. Therefore, a filesystem that is not mounted but is marked as dirty hasn't been properly unmounted and may be corrupted. Since we don't know what files were being written at the time of the crash, the entire filesystem will need to be examined to repair any problems. This can take a lot of time.

The ext3 filesystem adds a transactional journal to ext2. Even when mounted read-write, the system is marked as clean. However, if at boot time the journal contains information about incomplete writes, corruption may exist. In this case, since we know exactly which files were being written at the time of the crash (from the information in the journal), we can very quickly check and repair just the affected parts of the filesystem. Nonetheless, there are times when it's a good idea to perform an exhaustive check on an ext3 filesystem. It's possible for the hardware to introduce odd errors in rare circumstances. Some examples might be caused by gradual loss of power in a brownout, or severe vibration of the hard drive.

The fsck program is a front end to the standard filesystem checking programs on the system. The one for ext2/ext3, e2fsck, is used both to repair ext3 filesystems using the filesystem's journal and to exhaustively examine filesystems of either type.

If the root filesystem was not unmounted properly and has a journal, then when the kernel mounts the filesystem read-only, the kernel will read the journal and repair the filesystem. If rc.sysinit detects possible disk corruption, you'll be presented with a prompt that reads “Press Y within 5 seconds to force file system integrity check....”

Filesystem Recovery

- If / has journal, kernel examines it at boot
- **/etc/rc.d/rc.sysinit** runs **fsck** on filesystems marked in **/etc/fstab**
- **fsck** is a front end to other programs
- A failed **fsck** must be run manually



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

If you press **y**, then all filesystems marked for checking in **/etc/fstab** will be treated as if they were dirty, whether they're actually dirty or clean. This includes **ext3** filesystems. Normally it's okay to wait until the prompt times out.

Then, **rc.sysinit** will **fsck** the filesystems listed in **/etc/fstab** that have a positive integer in the sixth column. Any filesystems that have journals will use them to make rapid repairs, and any filesystems marked dirty (or being treated as such) will be exhaustively examined and repaired. If the filesystem is badly corrupted, **fsck** may fail and need to be re-run manually. You must provide the root password to get a shell from which you can run **fsck**, or the actual programs **fsck** runs; **e2fsck** (aka **fsck.ext2** and **fsck.ext3**), **dosfsck**, and so on.

e2fsck is used to repair both **ext2** and **ext3** filesystems. It should only be run against filesystems when they are not mounted or when they are mounted read-only. It takes as its argument the partition containing the filesystem to be checked. A number of switches are available, the most important of which are **-y**, which answers "yes" to all questions for which **e2fsck** would ordinarily prompt, and **-b block**, which instructs **e2fsck** to repair the filesystem using an alternate copy of the superblock. The appropriate location for the **block** argument may be determined using the **dumpe2fs** command. Because a corrupt filesystem may not be readable by **dumpe2fs**, it is advisable to have hard copy output of this information before a problem occurs.

Once filesystems have been repaired, it is possible to remount the root filesystem and bring it up from the **su** login state. However, it is preferable to type **exit**, which forces a complete reboot.

Sometimes the root filesystem is corrupted past the point where it is even mountable. Consequently, **/sbin/init** cannot be run. That forces us into the rescue mode described on the following page.

Recovery Run-levels

- Pass run-level to **init**
 - on boot from GRUB splash screen
 - from shell prompt using: **init** or **telinit**
- Runlevel **1**
 - Process `rc.sysinit` and `rc1.d` scripts
- Runlevel **s**, **S**, or **single**
 - Process only `rc.sysinit`
- **emergency**
 - Run **sulogin** only



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

In recovery situations, it is often helpful, (and depending on the problem possibly necessary) to boot to a run-level where less services are active. For example, consider if you have a service that causes the machine to panic each time it tries to start. In this case, the road to recovery starts by preventing the service from starting, so you can successfully boot the machine to a stable state and determine the problem with the service. The below listed run-levels are of particular importance in system recovery situations.

Run-level 1

Booting to run-level 1 will cause the system to process the `/etc/rc.sysinit` script followed by each of the `/etc/rc.d/init.d` scripts called in `/etc/rc1.d/*`. By default, RHEL will only call the single script in this runlevel, which after some basic checks and cleanup will exec `init S`.

Switching to run-level 1 from some other run-level (3, 5, etc.) is a convenient way to kill all daemons as each of the `/etc/rc1.d/*` kill scripts will be processed.

Run-level s, S, single

Booting to run-level single will cause the system to process the `/etc/rc.sysinit` script (if `/etc/inittab` is intact). If `/etc/inittab` is missing or corrupt, you can still boot to single mode, and in that case, you are given the root shell with no scripts processed.

Sometimes going to single user mode is overkill: interactive startup mode, invoked by typing “T” when “Welcome to Red Hat Enterprise Linux” appears at boot time, allows you to choose which services will run.

Run-level emergency

While technically not a run-level, emergency mode shares many characteristics of the above listed run-levels. You can only access emergency mode during boot by passing emergency as a parameter from the grub prompt. No scripts will be processed, and you are given a root shell.

Rescue Environment

- Required when root filesystem is unavailable
- Non-system specific
- Boot from CDROM (boot.iso or CD #1)
- Boot from diskboot.img on USB key



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

If the root filesystem is available and mountable, then you should be able to use it to fix problems that may occur. When it is not, then you must use a rescue environment. A rescue environment is a streamlined RHEL system that does not require the installed OS to run. Rather than working on the broken system itself, you work outside of the system in an environment that, while more limited than single user mode (or even sulogin mode), should provide enough tools to recover root.

There are several ways to boot into the rescue environment:

Boot from CDROM, then type **linux rescue** at the isolinux prompt

Boot from a diskboot .img USB drive, then type **linux rescue** at the prompt

Rescue Environment Utilities

- Disk Maintenance Utilities
- Networking Utilities
- Miscellaneous Utilities
- Logging: `/tmp/syslog` or `/tmp/anaconda.log`



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The rescue environment exists within a ramdisk image (referenced as `/dev/root`). Because of limitations on size and the number of inodes, many familiar utilities and device nodes are not available. However, tools related to disk maintenance (the probable reason for being in the rescue environment) and network connectivity are provided.

The following is an (incomplete) list of utilities provided by the rescue environment:

Disk Maintenance Utilities, including: a complete set of LVM utilities, for managing physical volumes, volume groups, and logical volumes; software RAID tools; swap commands; disk partition utilities; filesystem creators, checkers, debuggers, and labelers for `ext2`, `ext3`, `jfs`, `msdos`, `vfat`, and `reiser` filesystems.

Networking Utilities, including: network debuggers (**`ifconfig`**, **`route`**, **`traceroute`**, **`host`**) ; network connectivity tools (**`ftp`**, **`scp`**, **`ssh`**).

Miscellaneous Utilities including: shell commands (**`bash`**, **`chroot`**); process management tools (**`ps`**, **`kill`**, **`killall`**); editors (**`vi`**); `mt` tools commands; kernel module management commands; archiving and compression tools (**`dd`**, **`tar`**, **`cpio`**, **`gzip`**); **`rpm`**; file manipulation commands (**`cd`**, **`ls`**, **`mkdir`**, **`cp`**, **`mv`**, **`rm`**)#.

Within the rescue environment, system logging information can be found in the file `/tmp/syslog`. Booting information is in `/tmp/anaconda.log`. Some configuration files (`modprobe.conf`, `netinfo`, and device files (`[sh]da`, `loop0`) are located in `/tmp` as well.

Rescue Environment Details

- Filesystem reconstruction
 - Anaconda will ask if filesystems should be mounted
 - `/mnt/sysimage/*`
 - `/mnt/source`
 - `$PATH` includes hard drive's directories
- Filesystem nodes
 - System-specific device files provided
 - **mknod** knows major/minor #'s



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The rescue environment will attempt to reconstruct the harddisk's filesystem under the mount point `/mnt/sysimage`. Since the rescue environment is often used on systems with damaged or misconfigured filesystems, however, this operation might or might not work. A corrupted partition table will appear to hang the rescue environment (a shell with `fdisk` is available under *Alt-F2*, however.) Using **linux rescue nomount** as the boot prompt directive disables automatic mounting of filesystems and circumvents the hanging caused by bad partition tables. Careful inspection of the output of the `mount` command should determine the state of the reconstructed filesystem.

Because the standard installation provides nearly 7000 device nodes, administrators seldom need to create device nodes directly. In the rescue environment, device nodes are only provided for the most basic devices, including any fixed disks the kernel was able to autodetect.

In order to access any other devices, such as a floppy drive, the relevant device node must be created with `mknod`. Fortunately, the rescue environment's version of `mknod` automatically associates the appropriate device driver major/minor numbers with well-known device names. For example, the device node for the hard disk on the secondary IDE controller can be created with **`mknod /dev/hdc`**.

End of Unit 11

- Questions and Answers
- Summary
 - What are some things to check for
 - X problems?
 - Services problems?
 - Networking problems?
 - Boot problems?
 - How might you repair an ext2 filesystem?
 - What are some alternate boot methods?



For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Lab 11

System Rescue and Troubleshooting

Goal: To build skills in system rescue procedures.

Estimated Duration: 2 hours

Sequence 1: Repairing the MBR in the rescue environment

Scenario:

The rescue environment provides a last resort for repairing an unbootable machine, even when the bootloader or the root filesystem is damaged or misconfigured. In order to access the rescue environment, you will need either a `boot.iso` cdrom on a network that has the Red Hat installation tree (the `RedHat` directory) available via NFS, or a Red Hat Enterprise Linux CDROM.

Instructions:

1. Use the following command to overwrite the first stage of GRUB in your Master Boot Record with zeros. Specify the blocksize carefully. If you write too many zeros, you will overwrite your partition table as well, and this will become a much more difficult exercise. (Note that the command below assumes you are using IDE drives. You might need to modify the destination device.)

After typing the following command, check it three times and hit *Enter* but once.

```
# dd if=/dev/zero of=/dev/hda bs=446 count=1; reboot
```

Congratulations -- you have just wiped out your boot sector, but your primary partition table will still be intact. Attempt a reboot to confirm that your system is unbootable. Use the Red Hat rescue environment to repair the system.

Sequence 2: Installing software in rescue mode

Instructions:

1. Use the following command to overwrite the **mount** command.

```
# cp /bin/date /bin/mount
```

Congratulations -- you have just wiped out a key executable on your system. Upon attempting a reboot, you should find your system unbootable. Use the Red Hat rescue environment, along with its version of the rpm command and the library of RPMs provided by the installation tree, to repair the system. Hint: **/bin/mount** is part of the `util-linux` RPM.

Sequence 3: Troubleshooting Practice

System Setup:

1. Turn off iptables and mount the `/var/ftp/pub` directory from `server1` if it is not currently mounted.

```
service iptables stop
chkconfig iptables off
mkdir /mnt/server1
mount server1:/var/ftp/pub /mnt/server1
```

2. Install the Troubleshooting Practice RPM:

```
rpm -ihv /mnt/server1/gls/RPMS/rhce-ts-*
```

3. Ensure that your computer is configured as closely as possible to the following specifications:

- Authenticate users from your local `/etc/passwd` file. That is, do not run any network authentication scheme such as NIS or LDAP.
- Use 192.168.0.254 (`server1.example.com`) as your nameserver.
- Confirm that `/usr/local/bin` is part of your PATH environment variable.

The following items are required for some, but not all, troubleshooting problems. You may still do most problems if some of these items are missing.

- Change to runlevel 3, not runlevel 5. Confirm that the X server is not running (no `startx`). Only the local problems require this.
- Confirm that `/home` is a separate filesystem from the root filesystem and is local to the system (not an NFS mounted filesystem).

Instructions:

1. The Troubleshooting Practice problems come in three parts, each invoked by a separate command. The sections, commands, and number of problems in each section vary; therefore, run **`command count`**, where *command* is one of the following:

- For Local: **`tslocal count`** }
- For Networking: **`tsnetwork count`** }
- For Booting: **`tsboot count`** }

2. Invoke the first local problem by running:

```
# tslocal 1
```

This command will set up the problem and will explain the goal. The goal will be stored in the file `/etc/ts` for later reference. Spend three to eight minutes trying to solve the problem.

3. If you have not yet solved the problem, you may need a hint. Hints can be displayed by running the **tshint** command:

```
# tshint local 1 1
```

This will display the first hint for the first tslocal problem. Continue to invoke hints until you get enough information to solve the problem or until you run out of hints:

```
tshint local 1 2
tshint local 1 3
[ and so on ...]
```

The tshint command will tell you when you have reached the end of the hints. Again, do not spend more than five to ten additional minutes on this problem.

4. Whether or not you have solved the problem, run the tslesson command:

```
# tslesson local 1
```

This command will tell the lessons intended to be taught by the problem. Some tslesson messages also give step-by-step instructions on how to approach a particular problem.

5. If, after reading the hints and the lesson, you are unable to solve the problem, call the instructor for assistance.

6. Proceed with the remaining problems in the same way. e.g **tsnetwork 1** sets up the first network problem and **tshint network 1 1** shows the first hint for the first network problem.

Sequence 1 Solutions

1. Use the Red Hat rescue environment to repair the system.
 - a. Load the rescue environment by booting from a Red Hat installation media (either CDROM or PXE), and typing `linux rescue` at the boot prompt. Proceed with the normal installation defaults. Choose NFS image for the media type and use the following NFS information:
 - NFS server : `server1.example.com`
 - NFS directory : `/var/ftp/pub`
 - b. The rescue environment will ask if you wish to mount the hard drive's filesystems. Select **Continue** to mount the filesystems in read-write mode. Examine the output of `mount` to confirm that the filesystem was correctly reconstructed. You might want to refresh your memory by examining your disk's partitions with `fdisk` .
 - c. Note that your hard drive has been reconstructed under the mount point `/mnt/sysimage`. Examine `grub.conf` (on your hard drive) to confirm that it is appropriately configured.

```
# cat /mnt/sysimage/boot/grub/grub.conf
```
 - d. To reinstall GRUB, you must shift contexts, so that `grub-install` believes that the root of your filesystem is the `/mnt/sysimage` directory. Spawn a chrooted shell, run `grub-install`, and then exit.

```
# chroot /mnt/sysimage
# grub-install /dev/hda
# exit
```

(Or should the above fail to execute properly)

Exit the chroot environment and then type the command: **grub** at the bash prompt. This will place you into grub's command shell where you can enter the following commands:

```
grub> root (hd0,0)
grub> setup (hd0)
grub> quit
```
 - e. Now exit your rescue shell. Note that the rescue environment will unmount any partitions that you mounted. Eject the CD.

Sequence 2 Solutions

1. Use the Red Hat rescue environment, along with its version of the rpm command and the library of RPMs provided by the installation tree, to repair the system.
 - a. Load the rescue environment as in the previous exercise..
 - b. The rescue environment will attempt to automatically mount the hard drive's filesystems. Examine the output of mount to confirm that the filesystem was correctly reconstructed.
 - c. Verify the `util-linux` rpm on your harddrive, using a chrooted invocation of rpm. Do not forget to exit the chroot or the rpm installation will fail.

```
# chroot /mnt/sysimage
# rpm -V util-linux
# exit
```

- d. rpm should report that the `/bin/mount` executable has been modified. Reinstall the `util-linux` RPM from your installation tree (which has been NFS mounted under `/mnt/source`).

```
# cd /mnt/source/RedHat/RPMS
# rpm -ivh --force --root /mnt/sysimage util-linux*
```

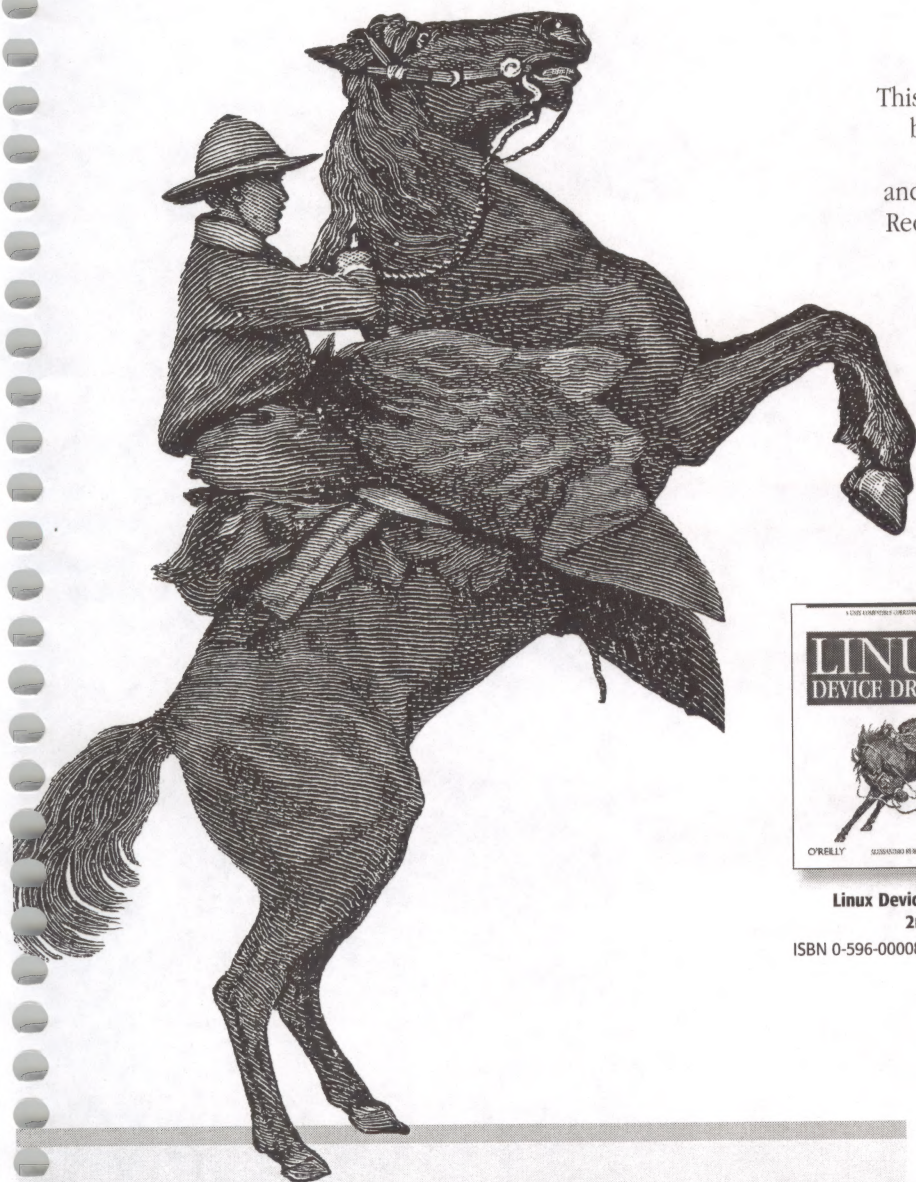
Note that the `util-linux` package was installed (the hash marks indicate this), although you may see some errors at the end of the process. As it turns out, this is harmless error, although in a production environment, you would want to test this out fully.

- e. Now exit your rescue shell. Note that the rescue environment will unmount any partitions that you mounted.

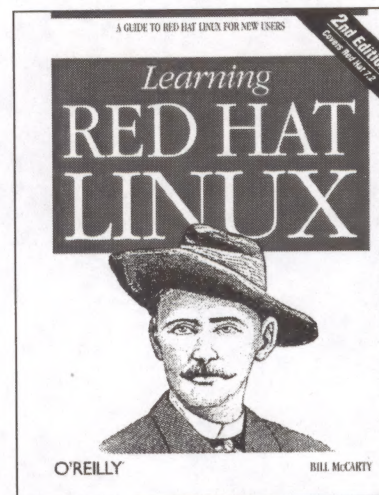
2126784

Saddle up with O'Reilly.

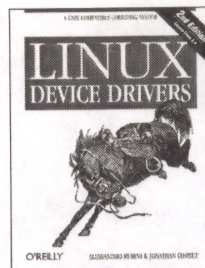
You've taken the reins on Linux, now learn how to use them with bestselling, award-winning books from O'Reilly. They're available from your favorite bookseller and accessible on the Web through SafariSM Tech Books Online.SM



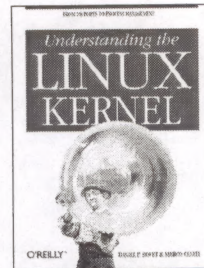
This new edition has been upgraded to cover installation and configuration of Red Hat Version 7.2.



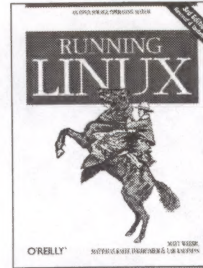
Learning Red Hat Linux, 2nd Edition
ISBN: 0-596-00071-5, \$34.95



Linux Device Drivers, 2nd Edition
ISBN 0-596-00008-1, \$39.95



Understanding the Linux Kernel
ISBN 0-596-00002-2, \$39.95

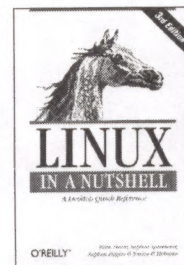


Running Linux, 3rd Edition
ISBN 1-56595-469-X, \$44.95

**Save 20%
on these titles**

**Order direct
from O'Reilly—
use code S2RHTS**

**800-998-9838
www.oreilly.com**



Linux in a Nutshell, 3rd Edition
ISBN 0-596-00025-1
\$34.95

SafariTM
TECH BOOKS ONLINESM

The Safari service lets you search, annotate, and read hundreds of the latest technical books from O'Reilly and other top tech publishers.

FOR A SPECIAL INTRODUCTORY OFFER VISIT:
<https://www.oreillynet.com/safaripromo/redhat-14.html>

O'REILLY[®]

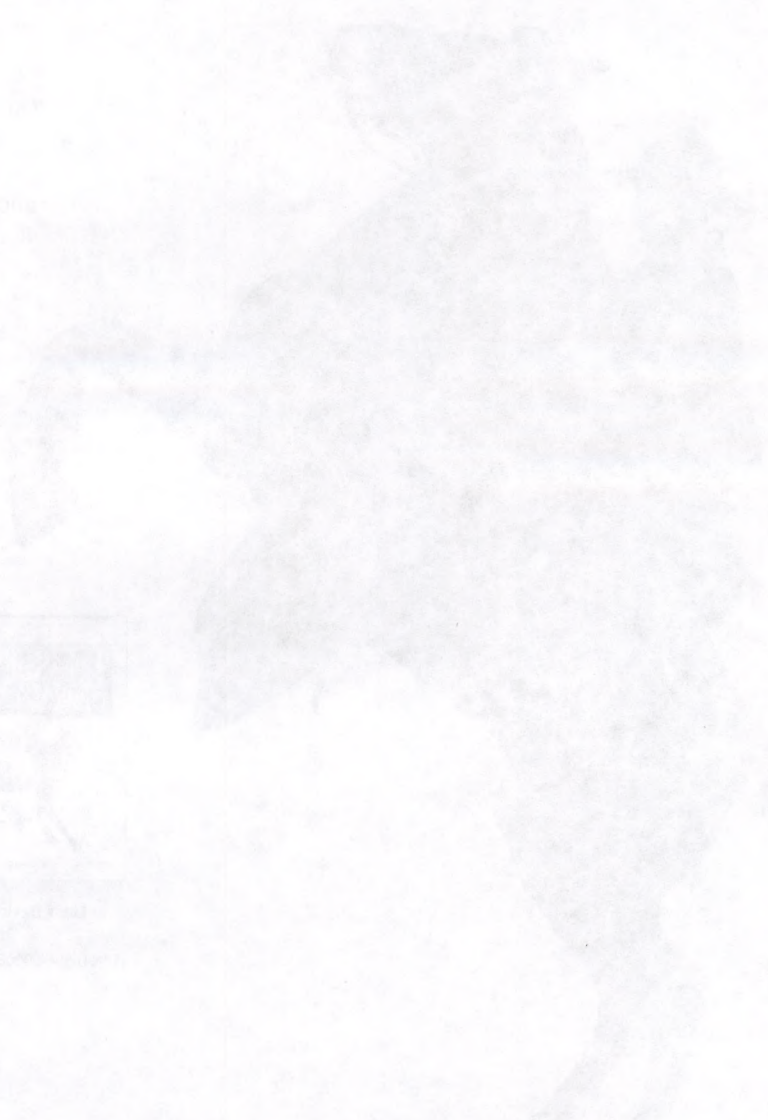
©2002 O'Reilly & Associates, Inc. O'Reilly is a registered trademark of O'Reilly & Associates, Inc. All other trademarks are property of their respective owners.

Saddle up with O'Reilly

It's time to saddle up with O'Reilly. We have the books you need to get the most out of your horse. From the basics of horse care to the advanced techniques of dressage, O'Reilly has it all. And now, with our new online service, you can get the books you need, when you need them, from the comfort of your home.



For more information on our books and online service, visit us at www.oreilly.com



Visit us at www.oreilly.com



Visit us at www.oreilly.com

Visit us at www.oreilly.com

Visit us at www.oreilly.com

Visit us at www.oreilly.com

Visit us at www.oreilly.com

Visit us at www.oreilly.com

Visit us at www.oreilly.com

O'REILLY

The O'Reilly online service is available at www.oreilly.com

Visit us at www.oreilly.com



Red Hat, Inc.
1801 Varsity Drive
Raleigh, NC 27606
PH: 919-754-3700
FX: 919-754-3713
<http://www.redhat.com/training>